T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KISIT PROGRAMLAMA İLE ÇİZELGELEME PROBLEMLERİNİN ÇÖZÜLMESİ

YÜKSEK LİSANS TEZİ

End.Müh. Erol ASLAN

Enstitü Anabilim Dalı	**:**	**ENDÜSTRİ MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Doç. Dr. Ayhan DEMİRİZ**

Mayıs 2010

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KISIT PROGRAMLAMA İLE ÇİZELGELEME PROBLEMLERİNİN ÇÖZÜLMESİ

YÜKSEK LİSANS TEZİ

End.Müh. Erol ASLAN

Enstitü Anabilim Dalı : ENDÜSTRİ MÜHENDİSLİĞİ

Bu tez 11 / 06 /2010 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Doç.Dr.Ayhan DEMİRİZ	Prof.Dr. Alpaslan FIĞLALI	Prof.Dr.Orhan TORKUL
Jüri Başkanı	Üye	Üye

ÖNSÖZ

Kısıt programlamanın çizelgeleme yaparken nasıl kullanıldığına ilişkin bilgilerin olduğu bu çalışma boyunca yardımlarını esirgemeyen saygıdeğer hocam Doç.Dr. Ayhan DEMİRİZ ‘e, aileme, çalışma arkadaşlarıma teşekkürlerimi sunarım.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

CHIP	: Constraint handling in prolog
CP	: Constraint programming
DSG	: Dallanma, sonuç çıkarma, daraltma
EDF	: Electricite De France
IBM	: International business machines
JRL	: Jet propulsion laboratory
KMP	: Kısıt mantıksal programlama
KOP	: Kısıt optimizasyon problemi
KP	: Kısıt programlama
KSP	: Kısıt sağlama problemleri
KTÇ	: Kısıt tabanlı çizelgeleme
MÖK	: Model öngörme kontrolü
MP	: Mantıksal programlama
MSG	: Minimum sabit güç
NP	: Nondeterministic polynomial
OPL	: Open programming language
SCIL	: Stanford center for innovations in learning
TGO	: Tümleşik gelişim ortamı
YA	: Yöneylem araştırması
YZ	: Yapay zeka

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Kısıt programlama, çizelgeleme, ILOG CP

Bu çalışmada, kısıt programlama ve kısıt programlama ile çizelgeleme problemlerinin çözümü detaylı bir şekilde irdelenmiştir. Matematiksel ve uygulamalı olarak konu açıklanmıştır. Uygulama bölümünde çeşitli kısıtları olan büyük ölçekli bir enerji yönetimi probleminin çözümü için IBM ILOG CP kullanılarak çizelgeleme ve minimum maliyetin bulunmasına çalışılmıştır. Problem çözümünden önce amaç fonksiyonu ve kısıtlar hem matematiksel hem de açıklamaları ile ifade edilmiş ve IBM ILOG CP ile adım adım çözüm yöntemi belirtilmiştir.

Sonuç olarak, ILOG CP kullanılarak bir çizelgeleme probleminin nasıl çözümlenebileceği detaylı bir şekilde ortaya konulmuştur.

SOLVING SCHEDULING PROBLEMS BY CONSTRAINT PROGRAMMING

SUMMARY

Key Words: Constraint programming, scheduling, ILOG-CP

In this study, constraint programming and solving scheduling problems by constraint programming were exhaustively examined. The issue is explained mathematically and practically. In practical section, it's been studied to find the scheduling and minimum cost by using IBM ILOG-CP for solving a large scale energy management problem with various constraints. Before solving the problem, both objective function and constraints are expressed both mathematically and with their explanations.

Consequently, it's been exhaustively asserted how a scheduling problem could be solved by using ILOG-CP.

BÖLÜM 1. GİRİŞ

Kısıt programlama, personel, makine ya da süreç adımlarına ilişkin gerçek dünyadaki zamanlama sorunlarının karmaşıklığını ele alır [1]. Çizelgeleme de bir kısıt sağlama ve optimizasyon problemidir. Bu nedenle kısıt-tabanlı çizelgelemede görevler; kaynak seçimi ve zamanlama değişkenleri, muhtemel öncelik kısıtlarla kısıtlanmış kaynak kısıtları, yönlendirme koşulları ve diğer koşullarla betimlenmektedir. Araştırma uzayı, kaynakların görevlere ve somut zamanlamaya karşı olası atamalarıdır. Görevler kaynaklara atanırken, zamanlama değişkenleri, sınırlı kapasiteler ve kurulum gecikmesi gibi kaynak kısıtlarıyla ayrıca kısıtlanmaktadırlar.

Bir çözüm yani bir çizelge, kaynaklar herhangi bir görevin üzerinde çalıştırılmak zorunda olduğunda nitelendiren kaynak ve zamanlama değişkenlerine bir atamadır. Bazen kısıt-sağlayıcı bir çözüm yeterlidir, fakat sıklıkla optimal bir çizelge istenmektedir. Bu durumda bir amaç fonksiyonu; çizelge uzunluğu, kaynak kullanımı ve son tarihlere ilişkin ortalama veya maksimum gecikmeler gibi eniyileme ölçütlerini tanımlamaktadır [2].

Modern kısıt çözücüler üç araştırma topluluğu tarafından irdelenmektedir [2, 3, 4]:

-Genel arama prosedürleri üzerine odaklı kısıt sağlama problemlerini (KSP'ler) araştıran (1) Yapay Zeka (YZ),

-Modelleme ve programlanabilirlik üzerine odaklı kısıt mantıksal programlama (KP/KMP) sonucunu doğuran (2) Mantıksal Programlama (MP),

-Kısıt problem formülasyonları için etkin algoritmalar üzerine odaklı (3) Yöneylem Araştırması (YA).

Kısıt-Tabanlı Çizelgeleme, yıllar geçtikçe KP'nın en başarılı kullanım alanlarından biri haline gelmiştir. Bu başarının kilit noktalarından biri, yöneylem araştırması (YA) ve yapay zeka (YZ) olarak adlandırılan çizelgelemeye dikkat eden iki eniyileme alanının bulmuş olduğu bir kombinasyon gerçeğine dayanmaktadır [2].

YA'daki dikkatin çoğu geleneksel olarak, daha çok basit matematiksel problemlere dayanan çizelgeleme problemleri üzerine odaklanmaktadır. Problemi kısa zamanda çözmek için, problemin karmaşık yapısını basit bir şekle indirgemektedir. Bir YA yaklaşımının, algoritmalarında yüksek seviyede verimlilik sağlamayı amaçladığını söylenebilmektedir. Bununla birlikte, uygulamalı bir çizelgeleme problemi bu klasik modeller kullanılarak modellendiğinde, genelde uygulamalı çizelgeleme durumunda var olan serbestlik dereceleri ve yan kısıtları atmaya zorlamaktadır. Serbestlik derecelerinin atılması, kullanılan çözümleme yöntemine bakılmaksızın ilginç sonuçların elenmesiyle sonuçlanabilmektedir. Yan kısıtların atılması, sadeleştirilmiş bir problem vermektedir ve bu sadeleştirilmiş problemin çözülmesi, orijinal problem için elverişsiz çözümlerle sonuçlanabilmektedir [5].

Bunun aksine YZ araştırmaları, daha genel çizelgeleme problemlerini inceleme eğilimindedir ve genel problem çözücü paradigmaları kullanarak problemleri çözmeye çalışmaktadır. YZ yaklaşımının, daha çok algoritmaların yazılım genelliği üzerine odaklandığını söyleyebiliriz. Bununla birlikte, bu şu anlama gelmektedir ki; YZ algoritmaları YA algoritmalarıyla karşılaştırıldığında, spesifik durumlar üzerinde zayıf bir şekilde işlem görmektedir. Bu nedenle bir yandan, daha sınırlı uygulama alanına sahip olmasıyla kıyaslandığında problemleri çözmek için etkin algoritmalar sunan YA kullanılmaktadır. Diğer taraftan, genel olarak daha fazla kabul edilebilir algoritmalar sağlayan fakat etkin bir YA algoritmasının bulunduğu spesifik durumlarda biraz zayıf performans sıkıntısı çeken YZ'ı kullanılmaktadır. İkisini birleştirmenin önemli yolu, YA algoritmalarını global kısıtlarla birleştirmeyle bulunmuştur. Bu gibi algoritmalar, bir global bakış açısındaki kısıtlar kümesini etkin

bir yolla hesaba katabilmektedir. Tipik çizelgeleme örneğinin global kısıtı, paylaşılan bir kaynaktaki kapasiteyi gerektiren tüm işlemlerin üzerinde çoğaltan bir kısıttır.

Kesinlikle alanın erken aşamalarındaki global kısıtlar içerisindeki çoğu algoritmanın temeli, YA'da bulunabilmektedir.

Lokalite prensibinin uygulanmasıyla [5,6] bu gibi özelleştirilmiş algoritmalar, geriye kalan kısıtlarla ilgilenen genel yayılım algoritmalarıyla yan yana çalışabilmektedir. Bu şekilde, biri KP'nın genel modelleme ve problem-çözücü paradigmasını sürdürebilmekte iken, etkin yayılım algoritmalarının tümlemesi yaklaşımın tüm performansını geliştirmektedir. Diğer bir şekilde işaret edilen, bir KP yaklaşımına tümlenmiş etkin YA algoritmaları, kullanıcının değişken bir çalışma alanındaki YA tekniklerinin verimliliğinden yararlanabilmesine olanak sağlamaktadır. Kısıt-tabanlı çizelgeleme alanı açısından baktığımızda bu durum :

-Kısıt sağlama problemleri (KSP'leri) olarak çizelgeleme problemlerinin doğal ve değişken modellemesini [2] ve

- Zaman ve kaynak kısıtlarının güçlü yayılımını ortaya çıkarmaktadır [5].

BÖLÜM 2. KISIT PROGRAMLAMA VE ÇİZELGELEME

2.1. Kısıt Programlama

Bir ayrık optimizasyon problemi, bildirimsel veya prosedürsel bir formülasyon olarak düşünülebilmektedir ve her ikisi de kendine özgü avantajlara sahiptir. Bildirimsel bir formülasyon, tamamen kısıtlara ve amaç fonksiyonuna odaklanmıştır. Algoritmik detaylar söz konusu olmaksızın aradığı çözümün ne tür bir çözüm olduğunun tanımlamasını sağlamaktadır. Prosedürsel formülasyon ise nasıl bir çözüm aranacağını ayrıntılarıyla belirlemekte ve bununla birlikte, birinin araştırmayı yönlendirmesi için problemdeki bilgilerden faydalanmasını sağlamaktadır. Tabii ki ideali, her iki evrenin en iyi öğelerini bir araya getirmektir ve kısıt programlamanın amacı budur.

Temelde birbirinden farklı görünen, bildirimsel formülasyon statik ve prosedürsel formülasyon dinamiktir. Örneğin; bir prosedürdeki bir noktaya x=0 koymak ve diğer bir noktaya x=1 koymak normal ve rutin olmaktadır fakat aynısını bildirimsel bir modelde yapmak, olursuz bir kısıt seti ile sonuçlanabilmektedir.

Bütün olumsuzluklara rağmen kısıt programlama mantığı, prosedürsel ve bildirimsel öğeleri bir arada toplayacak şekilde geliştirilmiştir. Fikirlerin gelişimi; mantıksal programlama, kısıtlı doyum, kısıt mantıksal programlama, eşzamanlı kısıt programlama, kısıtlı işlem kuralları ve kısıt programlama türleri üzerinden geçmiştir. Bu araştırma programından çıkan tek fikir; bir kısıt bir prosedürü çalıştırıyor gibi göstermektir. Bu, kısıt programlamanın ana fikridir [2].

2.1.1. Prosedürler olarak kısıtlar

Kısıt programlamacı, kısıtı bildirimsel olarak yazmaktadır fakat onu çözüm uzayı üzerinde çalışan bir prosedür gibi göstermektedir. Her bir kısıt, araştırma uzayını sınırlayan kısıtlama hafızasına katkıda bulunmaktadır. Kısıt hafızasındaki kısıtlar, kolay çözümleme oluşturulması bakımından anlaşılır olmalıdır. Kapsamlı çözümleme stratejisi, kısıt hafızasının çözümlemelerini sıralayarak orijinal problemin olası çözümlemesini bulmaktadır.

Uygulamada kısıt hafızası birincil olarak, bir değişkeni olası değerlerin tanım kümesi ile sınırlayan çok basit kısıtlar içermektedir. Bir değişkenin tanım kümesi genel anlamda, gerçek sayılar veya sonlu kümenin aralığıdır. Kısıt programlamaya kayda değer güç veren olay, değerlerin sayı olması gerekmemesi, herhangi bir türde amaçlar kümesi olabilmesidir.

Bir kısıtı bir prosedür olarak işleme sokma fikri, bilgisayar biliminde eğitim görmüş bir topluluk için çok normal bir şeydir çünkü bir bilgisayar programındaki komutlar tipik olarak prosedürleri çalıştırmaktadır [2]. Bu basit işleyiş, problem yapısından faydalanmak için güçlü bir araç sağlamaktadır. Çoğu pratik uygulamada, problemin bir bütün olarak sahip olmadığı özel yapıya sahip bazı altküme kısıtları bulunmaktadır. Var olan optimizasyon yöntemleri, örneğin; doğrusal bir parçayı ayırmak için Benders ayrıştırmayı kullanmak, bir ağ akış alt-problemini ön-çözümlemek vb. bu durumu bir yere kadar halledebilmektedir. Bununla birlikte, özel yapıdan faydalanan çoğu yöntem, tüm problemin yapıyı ortaya koymasına gerek duymaktadır. Kısıt programlama, altküme kısıtlamaları ile prosedürleri ilişkilendirerek bu zorluğun önüne geçmektedir. Bu, prosedürlerin kısıtların özelliklerinden faydalanmak için tasarlanmasını sağlamaktadır.

Durum derinlemesine irdelendiğinde, kısıt programlama, kısıt altkümelerinden ziyade bireysel kısıtlarla prosedürleri ilişkilendirmektedir fakat bu, global kısıtları kötü bir şekilde etkilemektedir. Global bir kısıt, yüksek yapılı kısıtlar kümesini temsil eden tek bir kısıttır. Bir değişken kümesinin belirgin değerleri almasını

gerektiren bir tüm farklı kısıt, örnek olabilmektedir. Bu, ikili bir eşitsizlik kümesini temsil etmektedir. Global bir kısıt belirgin bir yapı ile ilgilenmek için en iyi bilinen teknolojiyi çalıştırmak üzere dizayn edilebilmektedir.

Bu, çözücünün problemi bir farklılaşmamış kısıtlar kümesi olarak aldığı optimizasyonda kullanılan geleneksel yaklaşıma ters düşmektedir. Eğer çözücünün problemdeki herhangi bir altyapıdan faydalanması gerekiyorsa, ağ altyapısını bulan bazı ticari çözücüler gibi hareket etmesi gerekir. Bunun aksine global kısıtlar, kullanıcının özel yapıya sahip problemlerin bölümlerine karşı çözücüyü alarma geçirmesini sağlamaktadır.

Bireysel kısıtlara özel amaçlı prosedürler uygulanarak nasıl bir problem çözülebilir? Bu prosedürleri birbirine bağlayan nedir? Bu, kısıt belleğinin oyuna dahil olduğu yerdir. Her prosedür, değişken tanım bölgesindeki bazı değerleri elimine eden bir filtreleyici algoritma uygulamaktadır. Özellikle, bu kısıt için herhangi bir parçası olamayacak değerleri elimine etmektedir. Tanım bölgeleri, kısıtla belirtilen tanım bölgeleri içi kısıtlar ile geçerli olmaktadır. İşletilmek üzere bir sonraki kısıta devredilen kısıt belleğinin bir parçası haline gelmektedirler. Bu şekilde kısıt belleği, bir filtreleme prosedürünün sonuçlarını diğerlerine dağıtmaktadır.

Kısıtlar doğal olarak bir düzen içinde işletilmelidir ve bunu farklı yollardan farklı sistemler yapmaktadır. CHIP gibi kısıt lojik programlama sistemlerinde kısıtlar, bir lojik programlama diline (Prolog) tümleşiktir. ILOG Çözücü için yazılmış programlarda kısıtlar, kısıtların nasıl işletileceğini belirleyen C++ daki kodlama dizilimi gibidir. OPL Studio da yazılmış programlar, daha bildirimsel bir aramaya sahiptir ve sistem, işletim üzerinde daha fazla kontrol uygulamaktadır [2].

Bununla birlikte kısıtlı bir program, bir bilgisayar programı açısından yazılım olarak görülebilmektedir: kullanıcı nasıl yapılacağının detaylarını belirleyememesine rağmen komutlar prosedürleri başlatmaktadır ve kontrol bir komuttan diğerine geçmektedir. Bu, aslında bilgisayar programı olmayan fakat tamamen problemin bildirimsel komutları olan matematiksel programlama ile ters düşmektedir. Askeriyede lojistik programlamaya (planlama) karşılık George Dantzig'in önceki

doğrusal programlama uygulamasından dolayı programlar olarak adlandırılmaktadır [2]. Bu farklılığa rağmen kısıtlı bir programlama, kullanıcının kısıtları uygulamak için bir kod yazmaktansa kısıtları yazmasından dolayı bir bilgisayar programından ziyade daha çok matematiksel bir programlama gibi gözükme eğilimindedir.

2.1.2. Dallandır ve kes ile kısıt programlama arasındaki parallellik

Kısıt hafızasının orjinal probleme uygun olarak nasıl sıralanacağı sorun olmaktadır. Süreç, tamsayı programlama için dallandır ve kes algoritmalarına benzerdir. Problemin $D_1; \ldots ;D_n$ tanım bölgeleri ile $x = [x_1; \ldots ; x_n]$ değişkenleri içerdiğini farz edelim. Eğer D_j tanım bölgesi teke $\{v_j\}$ düşürülebilirse ve $v = [v_1; \ldots ; v_n]$ uygunsa, o halde x=v problemi çözmektedir. x=v düzenlemesi aslında kısıt hafızasını çözümlemekte ve kısıt hafızası orjinal problemde uygun hale gelmektedir. Bu, bir tamsayı programlama probleminin (kısıt hafızası) sürekli gevşemesinin çözümlenmesine ve tamsayı bir sonuç elde edilmesine benzerdir [2].

Kısıt programlama ile Dallandır-kes algoritmasının karşılaştırılması		
	Kısıt programlama	Dallandır-kes
Kısıt hafızası	Set içindeki alan kısıtları	Sürekli gevşeme (doğrusal eşitsizlikler)
Dallanma	Tekil olmayan bir alanın ayrılması veya bir kısıt üzerinde dallanma	Gevşeme çözümü olmayan bir tamsayı değeri üzerinde dallanır
Sonuç	Değişken etkiyi azaltır	Gevşeme için kesilmiş düzlem ekler
Sıçrama	------	Sürekli çözümlemeye bağlı kalmak içindir
Mümkün çözüm	Alan tekil olmalı ve kısıtlar uygun olmalıdır.	Gevşeme çözümün ayrılmaz bir parçasıdır
Çözümsüz düğüm	En az bir alan boşsa	Sürekli gevşeme mümkün değilse
Geriye dönük arama	Eğer düğüm çözümsüzse	Eğer düğüm çözümsüzse

Tablo 2.1. Kısıt programlama ile Dallandır-kes algoritmasının karşılaştırılması

Eğer tanım bölgeleri tamamen tekil değilse, o halde iki olasılık söz konusudur. Birinde, boş bir tanım bölgesi vardır ki bu durumda problem uygunsuzdur. Bu, dallandır ve kesteki uygunsuzluk sürekli bir gevşemeye benzerdir. İkinci olasılık şudur ki; bazı D_j tanım bölgeleri tek bir değerden fazlasını içermektedir ve bundan dolayı dallandırmayla kısıt hafızasının çözümlerini sıralamak gerekmektedir. İlki, her biri bir dala benzeyecek şekilde D_j'yi daha küçük tanım bölgelerine ayırarak x_j bölümlerine ayırabilmektir. Diğeri ise, teoride tüm çözümler sıralanana kadar dallandırmaya devam edebilmektir fakat dallandır ve kesteki gibi dallandırma ağacının her bir ağında yeni bir gevşeme (bu durumda, yeni bir tanım bölgesi kümesi) meydana gelmektedir. Gevşemeler, daha küçük olarak çalışmaya başladığından ve bundan dolayı kısıt yayılımı boyunca azaltıldığından dolayı bütün olarak ağaçta alçalmaya başlayarak daha sıkı hale gelmektedir. Arama, arama ağacının her yaprak düğümünde tanım bölgeleri tekil olana veya en az biri boşalana kadar devam etmektedir.

Bu süreç ve dallandır kes arasındaki temel paralel, hem dallandırma hem de sonuç çıkarmanın ikisini birden içermektedir. Kısıt programlama, bir kısıt hafızası (gevşeme) yaratmak için dallandırma ağacının her dalındaki iç tanım bölgesi kısıtlarından sonuç çıkarmaktadır. Dallandırma ve kesme, sürekli bir gevşeme oluşturmak için her ağdaki doğrusal eşitsizliklerden sonuç çıkarmaktadır. Sonraki durumda, gevşemedeki bazı eşitsizlikler, orjinal tamsayı programlama modelinin eşitsizlik kısıtları olarak görünmekte ve böylece sonuç çıkarmak için önemsiz olmaktadır ve diğerleri, gevşemeyi güçlendiren düzlemleri kesmektedir.

Hem kısıt programlamada hem de tamsayı programlamada meydana gelen sonuç çıkarmanın bir diğer şekli, aynı zamanda yararsız jenerasyon olarak da bilinen kısıta bağlı öğrenmedir. Yararsızlar, tipik olarak deneme bir sonuç (ya da kısmi bir sonuç) uygunsuz veya yetersiz bulunana kadar formüle edilmektedir. Arama devam ederken deneme sonucu ve belki de benzer nedenlerden dolayı memnun edici olmayan diğer sonuçları çıkarmak için tasarlanmış kısıtlardır. Yararsızlar, ana programın çözümü yetersiz veya uygunsuz bir çözüm ortaya çıkardığında aynı şekilde oluşturulan Benders kesmelerinin tamsayı programlama konseptine yakın bir şekilde

paraleldirler. Bütün olmayan bir sonucu kesmek için oluşturulmuş ayırma düzlemleri dışındaki kesme düzlemlerine daha az benzemektedirler [2].

Kısıt programlama ve tamsayı programlama, problem yapısını birincil olarak sonuç çıkarma safhasında kullanmaktadırlar. Örneğin, kısıt programlayıcılar global kısıtların yapısını kullanan filtrelerin dizaynı üzerinde oldukça fazla efor sarf ederken; tamsayı programlayıcılar güçlü kesme düzlemleri oluşturmak için belirgin problem sınıflarının çok düzlemli yapısı üzerinde çalışmaktadır.

İki yaklaşım arasında üç temel farklılık bulunmaktadır.

-Dallandırma ve kesme, uygun bir sonuçtan ziyade optimal bir sonuç aramaktadır. Bu ikincil bir farklılıktır çünkü, kısıt programlama çözücüsüne optimizasyon ilave etmek kolaydır. Basitçe, amaç fonksiyonun değeri üzerinde bir sınır empoze etmek ve uygun bir sonuç bulunur bulunmaz sınırı sıkılaştırmaktan ibarettir.

-Dallandırma ve kesme, küçük veya hiç kısıt yayılımı olmaksızın her ağdaki gevşemeyi çözmekte iken kısıt programlama daha çok yayılım üzerine odaklanmakta fakat bir gevşemeyi çözümlememektedir (Tanım bölgelerinin tekil hale geldiği özel durumda kısıt hafızasını çözdüğü söylenebilmektedir). Dallandırma ve kesmede gevşemenin çözülmesi, çoğunlukla arama ağacının budanmasını sağlayan optimal değer üzerinde bir sınır sağlamaktadır.

-Kısıt hafızası, dallandır ve kes yöntemleri bakımından çok daha zengindir çünkü basit bir şekilde iç tanım bölgesi kısıtlarından ziyade lineer eşitsizlikler içermektedir. Böylece, hibrit yöntemlerde iki tip kısıt hafızası eş zamanlı olarak kullanılabilmektedir [2].

2.1.3. Kısıt sağlama

Kısıt sağlama temelde, tamsayı programlamada dışbükey örtüye yaklaşık olarak paralel olan tutarlı bir kısıt kümesidir. Bu bağlamda; tutarlı, uygun veya tatmin edilebilir anlamına gelmemektedir. Kısıtlar, azalma miktarının korunan tutarlılık tipine bağlı olduğu geri izlemeyi düşürecek kadar belirgin olan uygun bir küme tanımı sağlamaktadır. Özellikle, güçlü bir n'li tutarlılık (n değişkenlerin sayısıdır) geri izlemeyi tümüyle elimine etmektedir ve tutarlılığın daha zayıf şekilleri belirli koşullar altında aynısını yapabilmektedir.

Eğer bir tamsayı/doğrusal programlama kısıt kümesi, dışbükey örtü ifadesi ise bazı bağlamlarda uygun kümenin belirgin bir ifadesini sağlamaktadır. Uygun kümenin dışbükey örtüsünün her yüzeyi açık bir şekilde belirtilmektedir. Bu şekilde, kümenin sürekli hafiflemesini çözerek problemi kolayca çözümleyebilmektedir. Dallandırma ve sınırlandırma ya da dallandırma ve kesme gibi bir geri izleme araması kullanmaya gerek kalmamaktadır.

Benzer şekilde, güçlü bir n'li tutarlılıktaki bir kısıt kümesi, problemin basit bir algoritmayla çözülmesini sağlamaktadır. Her değişken için, önceden yapılmış atamalara bağlı olarak hiçbir kısıtı bozmayan kendi tanım bölgesine ilk değer atanmaktadır (Bir kısıt, tüm değişkenleri atanana kadar bozulamaz.) Genelde, tanım bölgesi hiçbir değerin çalışmadığı bir noktaya ulaşılacaktır. Bu noktada geri izleme yapmak ve önceki atama için farklı değerler denemek gerekmektedir. Bununla birlikte, kısıt kümesi güçlü bir şekilde n'li tutarlılıkta ise, fırsatçı algoritma her zaman çalışmaktadır. Kısıt kümesi, uygun bir çözüm elde etmek için tamamlanamamış herhangi bir kısmi atamayı ortadan kaldıran kısıtlar içermektedir [2].

Yararlı olduğunu kanıtlamış tutarlılığın daha zayıf şekilleri, k-tutarlılık (k<n), ayrıt tutarlılık, genelleştirilmiş ayrıt tutarlılık ve sınır tutarlılığını içermektedir.

2.1.3.1. Hibrit yöntemler

Kısıt programlama ve optimizasyon, avantajlı olarak birleştirilebilen birbirini tamamlayıcı kuvvetlerdir. Problemler çoğunlukla, iyi bir şekilde yayılım yapan ve gevşeyen kısıtlara sahiptir. Hibrit bir yöntem, her iki türdeki kısıtı ele alabilmektedir. Kısıt programlamanın global kısıt düşüncesi, problemdeki alt yapıdan faydalanmakta iken yüksek yapılı problem sınıfı için optimizasyon yöntemleri gevşeme çözümlemeleri için yararlı olabilmektedir. Kısıt sağlama, global kısıtlar için filtrelenmiş algoritmalara katkıda bulunabilmekte iken, optimizasyon ise gevşemelere katkıda bulunmaktadır.

Hibridizasyonun, avantajlarından dolayı kısıt programlamanın işlem arama topluluğunda, izolasyonda kullanılan bir teknik olarak kurulmasından ziyade hibrit yöntemin bir parçası olarak kurulması olasıdır. En aşikar hibrit yöntem türü, kısıt çözücüler ile dallandır ve kes yöntemleri arasındaki paralellerden faydalanmaktır. Arama ağacının her ağında kısıt yayılımı, iç tanım bölgesi kısıtlarından bir kısıt hafızası yaratmakta ve çok düzlemli gevşeme de, eşitsizliklerden bir kısıt hafızası yaratmaktadır. Düşük tanım bölgesi değişkenler üzerindeki sınırları etkilediğinden ve değişkenler üzerindeki sınırlar tanım bölgelerini düşürebildiğinden dolayı iki kısıt hafızası birbirine değer katabilmektedir. Eşitsizlik gevşemesi, dallandır ve kes yöntemlerindeki gibi arama ağacını budayan optimal değer üzerindeki sınırı elde etmek için çözümlenmektedir. Bu yöntem, bir dallandırma, sonuç çıkarma ve gevşeme (DSG) yöntemi olarak adlandırılabilmektedir [2].

DSG yönteminin temel avantajı şudur ki; eşitsizlik formundaki bir problemi ifade etmeksizin çok düzlemli genişlemeden fayda elde edilmektedir. Eşitsizlik gevşemeleri, global kısıtlarla ilişkili gevşeme prosedürleriyle çözücü içerisinde genelleştirilebilmektedir, kullanıcının göremediği bir süreçtir. İkinci avantaj şudur ki; çözücüler en iyi bilinen gevşeme teknolojisini kolayca kullanabilmektedir. Örneğin; eğer global bir kısıt gezgin satıcı kısıtlarının bir kümesini ifade etmekteyse, problem için en iyi bilinen kesilmiş düzlemleri içeren doğrusal bir gevşeme

oluşturabilmektedir. Günümüzde, kesik düzlem teknolojisi kullanılmamaktadır. Çünkü genel amaçlı çözücülere uygulamak için sistematik hiçbir yol bulunmamaktadır. Bu problemin üstesinden gelmek için SCIL sistemi [21,2], kısıt programlamadaki global kısıt konseptini tamsayı programlamaya transfer etmektedir.

Hibrit yöntemlere gelecek vadeden diğer bir yaklaşım, genelleştirilmiş Benders dekompozisyonunu kullanmaktadır. [x,y] değişkenleri bölünmekte ve x'in değerleri üzerinden aranmaktadır. x için optimal bir değer bulma problemi, ana problemdir. Her sıralanmış v değeri için optimal bir y değeri, bu x=y varsayımına işletilmektedir; bu alt problemdir. Klasik Benders dekompozisyonunda, alt problem doğrusal veya doğrusal olmayan bir programlama problemidir ve çift yönlü çözümü, ana probleme ilave edilen bir Benders kesmesi getirmektedir. Benders kesmesi, ilerde sıralanmış tüm x değerlerinin v'den daha iyi olmasını gerektirmektedir. Daha fazla Benders kesmesi oluşturulamayana kadar Benders kesmeleri eklenmeye ve tekrar çözümleme yapmaya devam edilmektedir [2].

Bu proses, optimizasyon ve kısıt programlamayı birleştiren bir yolla genelleştirilebilmektedir. Alt problem, bir kısıt programlama problemi olarak kurulmaktadır. Çift yönlülük, klasik bir çift yönlülüğü genelleştiren çıkarım çift yönlülüğü olarak tanımlanabilmekte ve kısıt programlama yöntemiyle ilk olanın çözülmesi esnasında çözümlenebilmektedir. Çift yönlü çözüm, ana probleme eklenen genelleştirilmiş bir Benders kesmesi sağlamaktadır. Ana problem, karmaşık tamsayı programlama problemi gibi geleneksel bir optimizasyon problemi olarak formüle edilmekte ve çözümlenmektedir. Bu şekilde, dekompozisyon şeması, optimizasyon ve kısıt programlama yöntemlerini birleştirmektedir [2].

DSG ve genelleştirilmiş Benders dekompozisyonu, bir dizi problem kısıtlamalarını sıralayan ve her biri için bir genişlemeyi çözümleyen genel bir algoritmanın özel hali olarak gösterilebilmektedir. DSG'de, arama ağacının yaprak düğümleri kısıtlamalara denk gelmekte ve sürekli gevşemeler çözümlenmektedir. Benders'te ise, alt problemler problem kısıtlamalarıdır ve temel problem gevşemelerdir. Bu, tümlevsel optimizasyon, kısıt programlama ve yerel arama yöntemleri için genel bir şema zemini sağlamaktadır [22,2].

2.1.4. Performans konusu

Bir problem çözme teknolojisi, çözüm hızı olduğu kadar güç ve gelişme zamanını modellemek açısından da değerlendirilmelidir. Kısıt programlama, hata gidermek için matematiksel programlama modellerinden daha kolay olan kısa ve öz modellerle sonuçlanma eğiliminde olan esnek modelleme çalışma alanı sağlamaktadır. Buna ek olarak, yarı-prosedürsel yaklaşım, kullanıcının problemle en iyi nasıl uğraşacağına dair çözücü bilgisine ulaşmasını sağlamaktadır. Örneğin, kullanıcılar modeldeki alt yapıya işaret eden global kısıtları seçebilmekte ve model spesifikasyonu içinde aramayı rahatlıkla tanımlayabilmektedirler.

Kısıt programlama diğer avantajlara da sahiptir. İki alternatif formülasyondan birini seçmek yerine modelleyici, her ikisini de kolayca kullanabilmekte ve böyle yaparak çözümü kayda değer bir şekilde hızlandırabilmektedir. Modelleyici, çoğunlukla matematiksel programlamada olduğu gibi yapılandırılmış bir modele yan kısıtlar ekleyebilmektedir. Yan kısıtlar aslında, yayılımı geliştirerek çözüme yön verme eğilimindedir.

Diğer taraftan, modelleyici kısa ve öz bir model yazmak için büyük global kısıt veri sözlüğünü iyi bilmek zorunda iken tamsayı programlama modelleri sadece birkaç ilk terim kullanmaktadır. Etkin bir çözüm için doğru model ve arama stratejini bulmak adına birçok deney gerekli olabilmektedir ve proses bilimden çok sanattır [2].

Kısıt programlamanın tamsayı programlamaya ilişkin sayısal performansını özetlemek zordur. Kısıt programlama daha etkin yayılım gösterdiğinden dolayı sadece iki veya üç değişken içerdiğinde daha hızlı olabilmektedir. Kısıtlar pek çok değişkeni içerdiğinde tamsayı programlamanın sürekli gevşemesi kaçınılmaz hale gelebilmektedir [2].

Kabaca kısıt programlama, çizelgeleme problemleri için özellikle, kaynak kısıtlamalı çizelgeleme problemleri veya tamsayı programlama modelinin geniş olma

eğiliminde veya zayıf sürekli bir gevşemeye sahip olduğu diğer tümleşik problemler için daha etkili olabilmektedir. Eğer amaç uygun bir çözüm bulmak veya üretim süresi gibi min/max bir amacı optimize etmek ise bu kısmen doğrudur.

Tamsayı programlama, gezgin satıcı problemi gibi iyi çalışılmış bir çok düzlemliyi tanımlayan yapılandırılmış problemlerde üstün olabilmektedir. Kısıt programlama, gezgin satıcı problemi durumunda zaman penceresi gibi yan kısıtlarla karmaşık halde olduğunda veya daha büyük bir modelin parçası olduklarında rekabetçi hale gelebilmektedir.

Kısıt programlamanın tahminen kısıt yayılımı daha iyi olduğu için yüksek oranda-kısıtlanmış problemler için genellikle daha etkili olduğu söylenmektedir. Fakat, bir problem pek çok değişkenli bir maliyet fonksiyonu üzerine sıkı bir sınır koyularak yüksek oranda kısıtlanmış yapılabildiğinden dolayı bu yanıltıcı olabilmektedir. Böylesi bir manevranın kısıt programlama için problemi zorlu yapması olasıdır [2].

Kısıt programlama ve tamsayı programlamayı birleştirme durumunda bahsedilen avantajlara verilen önem iki yöntemin birbirini nasıl tamamlayacağı değişebilir olduğundan önemsiz olmaktadır. Tümlevlemenin sayısal avantajı önemlidir. Örneğin; son dönemlerde hibrit bir yöntem, ürün konfigürasyon problemlerini karmaşık tamsayı programlama (CPLEX) veya kısıt programlamadan (ILOG çözücü) 300-600 kez daha hızlı çözümlemiştir [23,2].

2.1.5. Kısıt problemleri

Bir kısıt sağlama problemi (KSP), bir n değişkenleri kümesi x_i (i = 1,…,n), her x_i değişkeni için kullanılabilir değerleri deklare eden bir aynı tanım bölgesi kümesi D_i (i = 1,…,n) ve değişken değerlerinin kullanılabilir kombinasyonlarını kısıtlayan değişkenler üzerindeki bir m kısıtlar kümesi $c_j(x_1,…,x_n)$ (j = 1,…,m) ile tanımlanmaktadır. Daha resmi olarak bir kısıt, $D_1 \times … \times D_n$ tanım bölgesi uzayının bir S altkümesini tanımlayan matematiksel bir bağlantıdır şöyle ki; eğer $(v_1,…,v_n) \in S$ ise kısıtın bir değişken ataması için sağlanmasını söylenmektedir $(x_1,…,x_n) = (v_1,…,v_n)$, $v_i \in D_i$ (i = 1,…,n). Bir değişken ataması, en az bir kısıt sağlanamazsa tutarsızdır. Kısıtlar topluluğu genellikle, kısıt hafızası olarak isimlendirilmektedir. KSP'ye karşı bir çözüm, tanım bölgesi değerlerinin kısıtları sağlayan değişkenlere atanmasıdır. Kısaca bir kısıt sağlama problemi aşağıdaki gibi tanımlanmaktadır [3].

Belirli D_i tanım bölgesi ve c_j kısıtları ile n değişkenleri x_i
bir çözüm bulmaktadır $(x_1,…,x_n) = (v_1,…,v_n)$
şöyle ki; $v_i \in D_i$ i=1,…,n
$c_j(v_1,…,v_n)$ j=1,…,m

değişkenlere ve kısıtlara detaylı bir şekilde göz atılmadan önce, bu tanımlama bir kısıt optimizasyon problemine genişletilebilir. Bir kısıt optimizasyon problemi (KOP), çözümler arasında bir öncelik kriteri belirleyen ilave bir amaç fonksiyonu $h(x_1,…,x_n)$ ile bir KSP olarak tanımlanmaktadır. Yani; eğer amaç h'yi minimize etmek ise o halde çözüm $(v_1,…,v_n)$, $h(v_1,…,v_n) < h(v_1',…,v_n')$ ise çözüm şeklinde belirlenmektedir. Dolayısıyla, bir $(v_1,…,v_n)$ çözümü, diğer tüm çözümler $(v_1',…,v_n')$ için $h(v_1,…,v_n) \leq h(v_1',…,v_n')$ ise optimaldir. Genelleme kaybı olmaksızın h'nin minimize edildiği varsayılmaktadır. Böylece kısıt optimizasyon problemi aşağıdaki gibi tanımlanmaktadır [3].

Belirli D_i tanım bölgesi, c_j kısıtları ve h amaç işlevi ile n değişkenleri x_i

şartıyla	$v_i \in D_i$	i=1,…,n
$c_j(v_1,\ldots,v_n)$	j=1,…,m	
minimal	$h(v_1,\ldots,v_n)$	
ile		
bir çözüm bulmaktadır	$(x_1,\ldots,x_n) = (v_1,\ldots,v_n)$.	

2.1.6. Kısıt problemlerinin modellenmesi

Değişkenler çeşitli tiplerde olabilmektedirler; örn: farklı tipte tanım bölgelerine sahip olmak:

-l alt sınır ve u üst sınır ile tamsayı etki alanları [l,u], örn: değişkenler bu aralıktaki herhangi bir değeri alabilmektedir;

-Lojik (veya Boolean) tanım bölgeleri [yanlış, doğru], genellikle [0,1] tanım bölgeleri ile tamsayı değişkenleriyle ifade edilmektedir;

-Bir seçim kümesini ifade eden sayma (veya seçimler) tanım bölgeleri, genellikle k'nın seçimlerin sayısı olduğu [1,k] aralıklı tamsayı değişkenleriyle ifade edilmektedir;

-Bu aralıktaki herhangi bir gerçek değeri alabilen alt ve üst sınır ile gerçek tanım bölgeleri;

-Her değerin bir küme olduğu küme tanım bölgeleri.

Aynı şekilde, her değişken tipi aşağıdaki gibi kendi kısıt tiplerine sahiptir:

-Tamsayı ve gerçek değişkenler için aritmetik kısıtlar (örn., $3x+y \leq z$);

-Lojik değişkenler için Boolean bağıntıları (örn: x∧ (y ∨ - z));

-Küme kısıtlar (öğe ve altküme bağıntılarını uygulamak için)

Tamsayı ve bağıntılı değişkenler üzerindeki kısıtlar, aynı zamanda sonlu tanım bölgesi kısıtları olarak adlandırılmaktadır. Genellikle aynı şekilde kullanılmaktadırlar ve YZ ve KP topluluklarındaki çoğu çözücünün baskın parçasını oluşturmaktadırlar. Çoğu çizelgeleme problemi, sonlu tanım bölgeli KOP'ler olarak ifade edilmektedir. Aynı zamanda, sonlu tanım bölgesi teknikleri, kontrol topluluğunda sürekli optimizasyon tekniklerinden daha az bilinmektedir. Kısıtların nasıl birleştirileceği ve türetileceğine dair kurallarla birlikte bir primitif kısıtlar kümesi (örn: tamsayı ifadeler üzerindeki eşitlik ve eşitsizlik bağıntıları), bir kısıt sistemi olarak adlandırılmaktadır. Kısıt sistemine bağlı olarak kısıtlar, ayrık veya sürekli değişkenler üzerinde lineer veya lineer olmayan olabilmekte ve eşitlik, farklılık, eşitsizlik (≠) ve özel amaçlı bağıntıları içermektedir [3].

Bununla birlikte daha jenerik olarak değişkenler, tanım bölgeleri ve kısıtlar, KP sistemi dışında ille de kademeli olmaksızın uygulamaya özgü tipte gelişigüzel olabilmektedirler. Aslında, çoğu kısıt programlama aracı, kısıt sistemlerine ilişkin olarak parametriktirler ve çoklu kısıt sistemlerinin paralel olarak kullanılmasına izin vermektedirler. Bu genelleme ve uzatılabilirlik, değişik tipte kısıtlarıyla çizelgeleme problemleri için KP'yi neyin bu kadar çekici yaptığının bir parçasıdır.

Bir amaç fonksiyonu, tek bir değişken kadar basit olabilmekte (örn: bir çizelgedeki son görevi temsil etmek) veya tek tek amaç fonksiyonun ağırlıklı toplamı gibi amaç kombinasyonlarını içeren daha kompleks kriterler biçimlendirebilmektedir [3].

2.1.7. Kısıt problemlerinin çözümlenmesi

KP'de sonlu tanım bölgeli KSP'lerin (ve KOP'ların) çözümlenmesinin en olağan yolu, bir tanım bölgesi indirgemesi, kısıt yayılımı ve arama kombinasyonudur. Bu teknikleri ve çeşitli örneklerini anlamak için, sonlu tanım bölgeli KSP'i devrelerin (mevcut tanım bölgeleriyle ifade edilen) ve eşiklerin değişkenler arasındaki kısıtlar olduğu bir grafik olarak düşünmek en iyisidir.

İşlem	**Kısıt grafiği**
x ve y'nin tanımlanması	x∈[0,10] y∈[0,10]
x<y-3 ekleme; yayılım	x≤y−3 x∈[0,7]——————— y∈[3,10]
x>3 ekleme; x düşürme; yayılım	x≤y−3 x∈[4,7] ——————— y∈[3,10] x≤y−3 x∈[4,7] ——————— y∈[7,10]

Tablo 2.2. Kısıt yayılım grafiği

Tanım bölgesi indirgemesi, kısıt yayılımı

Tanım bölgesi indirgemesi, tek terimli kısıt c(x)'in x değişkenine direkt olarak uygulanmasıdır. Örneğin; eğer x mevcut [0,10] tanım bölgesiyle bir tamsayı değişkeni ve c x>3 ise ardından x'in tanım bölgesi [4,10] olmaktadır [3].

Kısıt yayılımı, bir değişkenin kısıtlarla bağlantılı diğer değişkenlerinin tanım bölgesine karşılık tanım bölgesindeki değişimlerin yayılımıdır. Örneğin; x ve y mevcut [0,10] tanım bölgeleriyle tam sayı değişkenleri ise ve x≤y-3 kısıtı

eklenmişse, x'in alt sınırını hemen y'ye yayılabilmektedir, örn: y'nin tanım bölgesi [3,10] haline gelmektedir, ve y'nin üst sınırını x'e yayılabilmektedir, örn: x'in tanım bölgesi [0,7] haline gelmektedir. Bundan başka, x>3 kısıtı x'in tanım bölgesini [4,7]'ye düşürerek eklenirse, yeni alt sınır y'nin [7,0] haline gelen tanım bölgesine yaydırılmaktadır [3].

x≤y-3 gibi bir kısıtla, sadece x'in alt sınırı y'nin alt sınırına yaydırılabilmekte olduğuna dikkat edilmelidir. Yayılım, sıklıkla bir tanım bölgesi ifadesinin bileşenleriyle bağlanabilmekte veya indekslenebilmektedir. Bu gerçekleşim, özneller adı verilen, bir kısıt grafiğinde orjinal kısıtların yerine geçen etkin özelleştirilmiş eşiklerin başlangıcına yol açmaktadır [5]. Örneğin; x≤y-3 kısıtı iki öznel ile ifade edilmektedir lb(y) := max(lb(x)+3,lb(y)) ve ub(x) :=min(ub(y)–3,ub(x)) ki; lb ve ub sırasıyla tanım bölgelerinin alt ve üst sınırlarını göstermektedir. Örneğin, ilk öznel, x'in tanım bölgesinin alt sınırına bağlanmakta ve bu değer ne zaman değişirse yayılımı çalıştırmaktadır. Daha jenerik olarak, bir $c(x_1,\ldots,x_n)$ kısıtı, bir veya daha fazla "x_i in r_i" (i = 1,…,n) öznele dönüştürülebilmektedir ki; r_i, kısıttaki diğer değişkenler açısından x_i için mümkün değerleri belirlemektedir.

Bu kısıt yayılımı şekli, sözde ters tutarlılığa neden olmakta iken [6], grafikteki eşik (veya tersler) boyunca değişkenlerin tanım bölgesindeki tutarsız verileri kaldırmaktadır. Kendiliğinden kısıt yayılımı, tutarsız olan tüm olası değer kombinasyonlarını uzaklaştıramamasından dolayı kusurlu bir tekniktir. Örneğin; 7, her iki tanım bölgesinde de bulunmasına rağmen (x,y) = (7,7) yukarıdaki senaryoda tutarlı bir çözüm olmamaktadır.

Bu kusurluluk nedeniyle, tanım bölgesi indirgemesi ve kısıt yayılımı aramayla tamamlanmak zorundadır. Aramanın genellikle aralıklı kısıt yayılımı: arama süresince bir değişken değeri düzeltmek bu değişkendeki ileri yayılıma neden olabilmektedir. Örneğin; yukarıdaki kısıt problemi göz önüne alındığında eğer arama prosedürü x için değer olarak 6'yı seçerse yayılım sayesinde y'nin etki alanı [9,10] haline gelmektedir. Arama bakış açısından, tanım bölgesi indirgeme ve kısıt yayılımının amacı, arama öncesinde ve süresince arama uzayını olabildiğince daraltmaktır [3].

Arama

Kısıt programlamada arama yöntemleri, geliştirme tabanlı yöntemler ve onarma tabanlı yöntemler olmak üzere ayrılabilmektedir [7]. Geliştirme tabanlı yöntemler, kısıt programlamada aramanın en yaygın şeklidir. Değişkenlerin her biri, tam bir çözüm bulunana veya bir kısıt bozulana kadar bir değere artan şekilde atanmaktadır. Eğer bir kısıt bozulmazsa, son atama iptal edilmekte ve alternatif bir değer seçilmektedir. Eğer hiçbir değer ataması tutarlı değilse, arama önceden atanmış bir değere geri dönmektedir. Sonuç, bir derinlemesine aramadır. Akış şu şekilde ifade edilebilir.[3].

1.adım

$x = (x_1,\ldots,x_n)$; oluşturulması bu durumda x'de değişkenler bulunmaktadır.

2.adım

x'ten atanmamış bir x değişkeni seçilmelir.

3.adım

x'in etki alanından bir v değeri seçilmelir. Eğer x=v olması durumunda kısıt bozulursa geriye dönülür diğer türlü akış biter.

Bu akış, değişken seçimi ve değer seçimi olmak üzere birincil olarak iki seçim içermektedir. Değişken ve değerlerin seçildiği sıralama, arama etkinliği ve geri dönme ihtiyacını minimize etmeye çalışan bu seçimler için var olan çeşitli sezgiseller üzerinde kayda değer bir etkiye sahip olabilmektedir. Örneğin; mevcut en küçük tanım bölgeli değişkenleri seçen sezgisellerin ve yayılım esnasında diğer değişkenlerin asgari tanım bölgelerini azaltan değerlerin pek çok uygulamada iyi performansa neden oldukları bilinmektedir [8,3].

Tanım bölgesinden bağımsız bu gibi sezgiseller etkin olabilmekte iken tanım bölgesine özgü sezgiseller daha da iyi olmaktadır. Bu, problem (örn: tipik görevler sırası) hakkında ilk bilginin bir sezgisel (örn: zaman sırasına göre değişkenleri ve değerleri seçmek ve maksimum toplam çizelge uzunluğunu kısıtlamak) olarak kullanılabildiği çizelgeleme problemleri bağlamıyla ilgili olmaktadır [9,3].

Diğer seçim, geri izleme prosedüründedir. Kronolojik geri izleme, atamanın ters sırasındaki önceki değer ve değişken seçimlerine geri izleme ve bozma, gerçekleştirmesi en kolayıdır, çoğunlukla oldukça etkilidir ve böylece kısıt programlamada kullanılan en yaygın şekildir. Bununla birlikte, KSP topluluğunda akıllı geri izlemenin çeşitli şekilleri keşfedilmiştir. Örneğin; kolay problemlerin bazen, algoritmaların arama süresinin çok sonrasına kadar belirlenmeyecek yanlış seçimleri önceden yapmasına neden olmaktadır [10,3]. Bu durumda, arama ağacının en altında başarısız arama üzerinde zaman harcamak yerine, bir geri sıçrama algoritması en baştaki tutarsız atamayı tanımlamaya çalışacak ve direkt olarak bu değişkene doğru geri izleme yapacaktır.

Aramada gömülü farklı ters tutarlılık dereceleri, hiçbir yayılım işletilmemişse saf oluşturur ve sınamadan tüm ağ tutarlılığı uygulanmışsa tam ileri bakmaya kadar farklı ileri bakma derecelerine neden olmaktadır [11,12,3]. Tam ileri bakmanın işletimi masraflı olabildiği ve geri izlemede kaydettiğinden daha fazlasını destek işlemine eklediğinden dolayı kısmi ters tutarlılık ile geri izleme araması, çeşitli kısıt problemleri için en iyi işletim göstermiştir [3].

Eğer kısıt problemi bir optimizasyon problemi ise geliştirme tabanlı arama, her defasında yeni bir sonuç $(v_1,\ldots,v_n)$ bulunduğunda yeni bir kısıt $h(x_1,\ldots,x_n) < h(v_1,\ldots,v_n)$ ekleyen bir mekanizmayla kolay bir şekilde çoğaltılabilmektedir. Bu, sonraki çözümlerin artan şekilde daha iyi amaç değerlerine sahip olmalarını çabuklaştırmakta ve arama ağacının parçalarının kaldırılmasında çok etkili olabilmektedir. Sonunda, son bulunan çözüm optimal çözüm olarak geri bildirilmektedir. Bu tekniğin çizelgeleme problemlerinde etkin olan bir varyasyonu, l ve u alt ve üst sınırlarını $h(x_1,\ldots,x_n)$ üzerinde dereceli olarak daha dar tutan ikili aramadır. Algoritma ilk olarak $h(x_1,\ldots,x_n) \leq (l+u)/2$ kısıtını eklemektedir. Eğer bir

sonuç $(v_1,\ldots,v_n)$ bulunursa u, $h(v_1,\ldots,v_n)$'ye ayarlanmakta aksi durumda l, (l+u)/2+1'ye ayarlanmaktadır. l veya u için yeni değer, $h(x_1,\ldots,x_n) \leq (l+u)/2$ amaç sınırında yerine koyulmakta ve arama tekrar başlatılmaktadır. Bu, bir sonuç bulunduğunda $l \geq u$'ya kadar tekrarlanmaktadır. Gene de diğer bir etkili ve popüler varyant, $h(x_1,\ldots,x_n) \leq l$ kısıtındaki h üzerindeki l limitinin bir alt sınırdan başladığı döngülü derinleşmedir ve bir sonuç bulunana kadar yükseltilmektedir. Eğer ilk l optimuma yakınsa bu etkilidir. Büyük problemler için çok etkili olduğunu kanıtlamış olan daha güncel bir teknik, seçilen değer sezgiselin birkaç hata yapacağını varsayan ve bundan dolayı ilk olarak sezgiseldeki izinli sapmaların sayısını sınırlandıran sınırlı farklılık aramadır [13,14,3].

Onarma tabanlı yöntemler, değişkenlere tam bir atamayla başlamaktadır. Eğer bu atama tutarsız ise, örn: en az bir kısıt bozulmuşsa, atama bir sonuç bulunana kadar değişkenlerin bir veya daha fazlasına farklı değerler atanarak tekrarlanarak onarılmaktadır. Akış şu şekilde ifade edilebilir [3].

1.adım

v tutarsız ise;

2.adım

$(x_1,\ldots,x_n)$ için ilk sonuç için $v = (v_1,\ldots,v_n)$ oluşturulmalıdır.

3.adım

v'den tutarsız bir x=v ataması durumunda x iiçn yeni v´ değeri seçilmelidir.

4.adım

v'de x= v´ atanmalıdır ve akış bitirilmelidir.

Bu akışta da değişken seçimi ve değer seçimi olmak üzere iki seçim içermektedir. İyi bilinen sezgiseller, uyumsuzlukların sayısını düşüren değerlerle en büyük sayıdaki

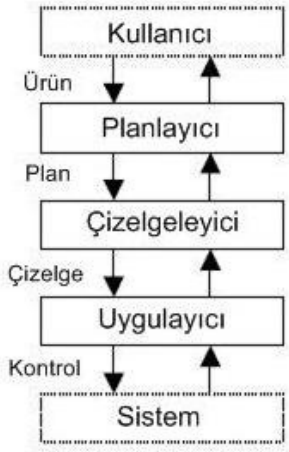
kısıtları bozan değişkenlerin seçimini içermektedir. Bir optimizasyon problemi durumunda onarma tabanlı yöntemler, tepe tırmanma veya benzetilmiş tavlama gibi iyi bilinen optimizasyon teknikleriyle kolay bir şekilde birleştirilebilmektedir. Bu gibi durumlarda değişken ve değer seçimi yalnızca kısıt uyumsuzluğundaki azalışı hesaba katmakla kalmamakta aynı zamanda onarmak için değer ve değişkenler seçildiğinde amaç fonksiyonundaki düşüşü de hesaba katmaktadır [3].

Geliştirme tabanlı yöntemlerin aksi olarak onarma tabanlı yöntemler, tipik olarak tamamlanmamaktadırlar, örn: global optimumu veya tüm kısıtları karşılayan bir değişken atamasını bile bulmaları garanti değildir. Bundan dolayı, onarma tabanlı yöntemler tipik olarak, onarma sayısı üzerindeki bir üst sınır gibi ilave sonlandırma kriterleri gerektirmektedir.

Her iki arama algoritması sınıfı da, rastgeleleştirme teknikleriyle ayrıca çoğaltılabilmektedir. Daha önceden belirtildiği gibi, özellikle geliştirme tabanlı arama, pek çok çözümün var olduğu ve amaç üzerindeki hiçbir güçlü üst sınırın yayılım boyunca yardımcı olmadığı kolay optimizasyon problemleri için verimsiz olabilmektedir. Geri izleme üzerinde rastgele yeniden başlatmalar ve sınırlar kullanan benzer tekniklerin bir dizi çizelgeleme problemleri için etkili olduğu gösterilmiştir [15,3].

2.2. Çizelgeleme

Çizelgelemenin geleneksel konumu, planlama ve yürütme arasındadır: arzu edilen bir ürünler kümesi için; bir planlayıcı ürünleri dağıtacak olan görevlerin sırasını belirlemekte, çizelge, spesifik kaynaklara, işlemlere ve görevleri işletmek için zamanlamalarına karar vermekte ve ardından bir uygulayıcı, ürünlerin üretildiği özelleştirilmiş zamanlardaki işlemleri yapmak için sistemin kaynaklarını yönlendirmektedir (Şekil 2.1.) [3].



Şekil 2.1. Bağlam içi çizelgeleme [3]

Dahil olan terimler kapsamlıdır. Ürünler, fiziksel ürünler olabildiği gibi aynı zamanda, bir üniversite ders programlama problemindeki dersler, bir kadro çizelgeleme problemindeki kadro atamaları veya bir lojistik problemindeki madde lokasyonları da olabilmektedir. Kaynaklar, makineler veya makine bileşenleri, insanlar veya yararlı bir işlemi uygulayabilecek herhangi bir nesne olabilmektedir. Görevler, bir ürünün taşınması veya değiştirilmesi gibi sistemlerde belirtilmemiş kaynaklarda var olan tek adımlardır. İşlemler, özel görevleri işleten kaynakların spesifik kapasitelerinden bahsetmektedir. Sistem, basit bir makineden bir makineler fabrikasına, bir insan, fabrika, araç vb. organizasyonuna kadar kaynaklar topluluğudur.

Şekil 2.1.'in kurulumunda, karmaşıklık aşağıdan yukarıya doğru artmaktadır ve bu nedenle daha yüksek seviyelerdeki uygulamalar, genellikle daha az otomatikleştirilmektedir (örn: daha fazla insan aracılığı veya en azından hazırlığı). Örneğin; çoğu uygulamada çizelgeleme otomatikleştirilse bile, planlama hala elle yapılmakta veya ürünlerden planlara kadar basit bir haritalandırma sağlayan uzmanlar tarafından hazırlanan bir planlar kütüphanesine bel bağlamaktadır. Çizelgeler için bile, çizelgelerin insanlar tarafından okunabilirliği ve değiştirilebilirliği, var olan teknolojinin sınırlamaları nedeniyle zaruri bir gerekliliktir [3].

Bununla birlikte belirgin veri tabanlarında, hem planlama hem de çizelgeleme tamamen otomatikleştirilebilmekte ve böylece tümleşik bir kontrol sisteme yerleştirilebilmektedirler. Buna bir örnek, modern, dinamik olarak tekrar konfigüre edilebilir, çok fonksiyonlu bir baskı makinesinde (fotokopi, yazıcı, faks makinesi vb.) kağıt nakli ve baskı işlemlerinin planlama ve çizelgelemesidir. Burada, planlama ve çizelgeleme makinenin sistem kontrol yazılımının parçasıdırlar ve çizelgeleme, sistemin kontrolcü hiyerarşisinde başka bir seviye olarak düşünülebilmektedir. Çizelgeleme daha yüksek seviyede veya toplam işlemleri düşünmekle meşgul olmakta ve daha uzun vadeli bir tablo çizmektedir ve bundan dolayı çizelgeleme problemleri, genelde NP zor olan kesinlikle kompleks ve birleşimsel problemlerdir [16,17,3]. Aynı zamanda, gömülü çizelgelerin sistemin uygulanmasıyla paralel (çevrimiçi çizelgeleme) ve bir geri bildirim döngüsünde (reaktif çizelgeleme) işletilmesi ve zor gerçek-zamanlı sınırlar içinde sonuçlar sağlanması gibi daha düşük seviyedeki kontrolcülerin genel özelliklerini taşıması beklenmektedir.

Klasik çizelgeleme problemleri genellikle, akış tipi, atölye tipi veya sipariş tipi problemleri makine atölyesi benzetmesi kullanılarak kategorize edilebilmektedir [18,3]. Bir makine atölyesi, her biri tek seferde bir görev üzerinde çalışabilen bir makineler kümesinden oluşmaktadır. İşler, her biri uygulanması süresince bir kaynak gerektiren görevlerden oluşmaktadır. Özel makine atölyesi problemleri, bir işteki görevlerin sırasının kısıtlanıp kısıtlanmadığı ve kaynak kullanımına karar verilip verilmediği konusunda farklılık göstermektedir. Belirleyici algoritmalar, sadece birkaç (2-3) makineli ve işli problemler için kullanılmaktadır.

Kaynaklar, işler ve görevler benzetmesi genellikle çoğu çizelgeleme problemine uygulanabilmesine rağmen, gerçek hayattaki problemler öncelik ve kaynak kısıtlarından tipik olarak çok daha geniş çeşitlilikte kısıtlara sahiptir [19,3]. Kısıt-tabanlı yöntemler, problemler zor olduğunda, tanım bölgesine özgü ve gereksiz kısıtlar var olduğunda ve problemler sıklıkla değiştiğinde başarılı olduğunu kanıtlamıştır [20,3].

Bir örnek olarak, basit fakat tam bir atölye tipi üretim çizelgeleme problemi bir KOP olarak tanımlanmaktadır. Verilenler, bir iş kümesi ve bir makineler kümesidir. Her bir iş, atanan makineler üzerinde belirli bir düzen içinde işletilmek üzere belirli süreklilikte bir görevler kümesinden oluşmaktadır. Her makine, bir seferde yalnızca bir görevi işletebilmektedir. Problem, bir çizelge bulmaktır; örn: üretim süresini minimize eden her bir görev için bir başlangıç zamanı.

Bir görevi, t.s ve t.d'nin görevin başlangıç ve bitiş zamanlarını temsil ettiği bir t aralık değişkeni ile ifade edebilmekteyiz ve t.e = t.s+t.d bitiş zamanıdır. Ayrıca bir makineyi, tekil bir r kaynak değişkeni ile ifade edebilmekteyiz. Tüm zamanların tamsayı olduğunu farz edilmelidir. Son olarak, üretim süresi l değişkeni ile ifade edilmektedir. Ardından aşağıdaki kısıtlar uygulanmaktadır:

Her t görev için: t: $0 \leq t \leq l$.

Belirli d sürekliliğinde her t görev için: d: t.d = d

$(t_1,\ldots,t_n)$ görevli her iş için: $t_i \leq t_i+1$ $(i = 1,\ldots,n-1)$.

Belirli atanmış r makinesi, her t görev için: ayır (r,t).

Amaç fonksiyonu h = l olarak tanımlanmaktadır; örn: amaç minimizasyonu sağlamaktır. Belirli bir uygun kaynak kısıt sistemi, ayrılmış T_r görevleriyle tek r kaynağı otomatik olarak T_r'deki tüm t_i, t_j için $t_i \leq t_j \vee t_j \leq t_i$ kısıtlarını uygulamaktadır. Alternatif olarak, global kısıtlar kullanılarak kaynak kısıtları ayrılmış T_r görevler listesiyle her r kaynağı için kümülatif (T_r, [1,1,…], 1) olarak ifade edilebilmektedir.

Geleneksel olarak, çevrimdışı/öngörücü çizelgeleme, bu problem kısıtların kısıt çözücüye atanması ve ardından arama prosedürünü çağırmakla çözülmektedir; örn: (X,l)'yi minimize etmek ki, X, tüm t görev aralıkları ve l değişkeninin listesidir. Bu, yukarıda açıklanan yayılma ve arama tekniklerini kullanarak X'teki değişkenleri somutlaştıracaktır.

Çevrimiçi çizelgelemede, görevler ve kısıtları aşamalı olarak açığa çıkabilmekte ve çizelge, önceki çizelgenin işletimine karşılık eş zamanlı çalışabilmektedir.

Bundan başka kısıtlar (örn: kaynakların kullanılabilirliği), problemin bir parçasının veya tümünün tekrar çizelgelenmesinin gerektiği durumlarda çizelgeleme ve/veya işletim süresince değişebilmektedir.

Çevrimiçi çizelgelemeye en basit yaklaşım, çizelgeleme problemini bir kısıt problemleri dizgesi olarak ele almak ve bağlılıkları bir problemden diğerine transfer etmektir. Bununla birlikte, bu yaklaşıma karşılık optimizasyonlar olasıdır ve gerçek zamanlı uygulamalar için gereklidir [3].

Daha genel olarak, çizelgeleme bir kontrol sistemine gömüldüğünde kısıt çözücüyü bu çevreye adapte etme gereksinimi gibi benzersiz gereksinimler kümesi ortaya çıkmaktadır. Bu sorunlar; bellek yönetimi, kısıtların güncellenmesi ve değişkenlerin atanmasında gerçek zamanlı sorunlar ve kontrol uygulamasına karşılık ara yüzü içermektedir.

Endüstriyel kısıt tabanlı uygulamalar için yayınlanmış pek çok örnek bulunmaktadır. Tümleşik bir kontrol sistemine gömülü dağıtılan gerçek zamanlı çizelgeler için örnekler; Xerox'un baskı makinesi kağıt yolu çizelgesi [17], NASA'nın Remote Agent planlayıcısı (RAX-PS) [24] ve çizelgeleyicisi ve JPL'nin Aspen planlayıcısı ve çizelgeleyicisini içermektedir. NASA Ames' Remote Agent, deneysel uzay gemisi Deep Space One'ın parçası için RAX-PS sistemi, çevrimiçi kısıt yönetimi için özelleştirilmiş derinlimesine arama stratejileri ve hızlı işlemleri olan bir kısıt çözücü ile desteklenmektedir. JRL'nin NASA uzay gemisi için Aspen planlayıcı ve çizelgeleyicisi, çevrimiçi kısıt yönetimi sistemi içermekte ve yinelemeli onarma tekniklerini kullanarak gerçek zamanlı karşılık sağlamaktadır [3].

2.2.1. Karmaşıklık

Çoğu gerçek çizelgeleme problemi maalesef NP zordur [16,17,3]. Çizelgeleme problemleri bazen kesin varsayımların hafifletilmesiyle, örn: görevleri önceden

ayırmayı sağlayarak, kolaylaştırılabilmektedir. Bununla birlikte, çoğu araştırma jenerik sezgiseller bulmanın ve programcılara etki alanına özgü bilgiyle algoritmalara yol göstermenin altını çizmektedir. Bundan dolayı, sadece kısıt sistemlerine ilişkin olarak değil yayılma kurallarına ilişkin olarak da parametrik olan beyaz kutu kısıt çözücülere yönelim başlamıştır [5,3]. Spesifik gerçek zamanlı uygulamalar için özel karmaşıklık analizi uygulamak yararlı olmaktadır.

2.3. Kısıt-tabanlı Çizelgeleme

2.3.1. Çizelgeme için kısıt programlama modelleri

Kaynaklar, zaman içerisinde herhangi bir noktada aşılamayan belirli bir kapasiteye sahiptir. Ayrıca, işlemler ve amaç işlevi arasında geçici kısıtlar bulunabilmektedir. Problemin çözülmesi, hem geçici hem de kaynak kısıtlarına uyarak amaç fonksiyonunu optimize etmek için her işlemi ne zaman devreye sokacağına karar vermektir[1].

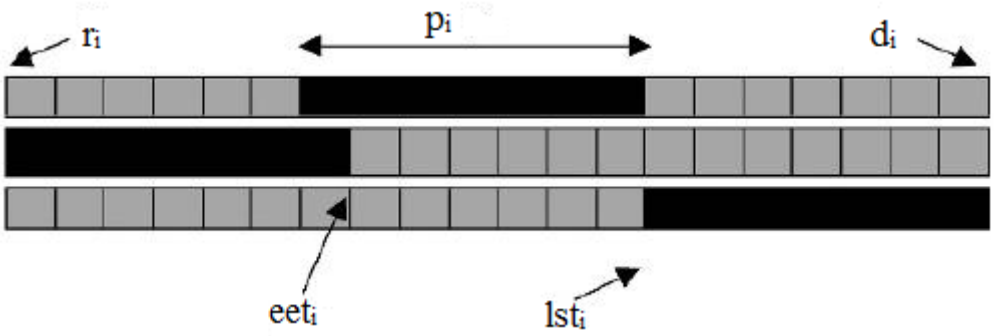
2.3.1.1. İşlemler

Bir problem içindeki işlem türlerine bakıldığında, sonsuz öncelikli olmayan çizelgeleme, sonsuz öncelikli çizelgeleme ve elastik çizelgeleme fark edilmektedir. Sonsuz öncelikli olmayan çizelgelemede işlemler kesintiye uğratılamaz. Her işlem, başlangıç zamanından bitiş zamanına kadar kesinti olmaksızın işletilmelidir. Sonsuz öncelikli çizelgelemede işlemler, bazı diğer işlemlerin işletilmesine izin vermek gibi herhangi bir zamanda kesintiye uğratılabilmektedir. Elastik çizelgelemede, bir işleme atanmış kaynak miktarı A_i, atanmış kapasite üzerindeki zamanların toplamının enerji

olarak adlandırılan belirli bir değere eşitlenmesini sağlayarak herhangi bir t zamanında 0 ve kaynak kapasitesi arasındaki herhangi bir değeri alabilmektedir. Sonsuz öncelikli olmayan bir işlem durumundaki eşdeğer enerji kavramı, işletim zamanının ve gereken kapasitenin ürünüdür[1].

Sonsuz öncelikli olmayan bir çizelgeleme problemi aynı şekilde bir KSP olarak etkin bir şekilde şifrelenebilmektedir. Her işlem için üç değişken ortaya koyulmaktadır, start(A_i), end(A_i) ve proc(A_i). Sırasıyla A_i'nin başlangıç, bitiş ve işlem zamanını temsil etmektedirler[1].

Çizelgeleme probleminin ilk verisinde tanımlanmış olan A_i aktivitesinin r_i serbest bırakma tarihi ve d_i son tarihi ile [r_i, d_i], A_i'nin işletilmesi gerektiği zaman penceresidir. Buna bağlı olarak, start(A_i) ve end(A_i)'nin ilk etki alanları sırasıyla [r_i, lst_i] ve [eet_i, d_i]'dir. Buradaki lst_i ve eet_i, A_i'nin en geç başlama tarihi ve en erken bitiş tarihi anlamına gelmektedir. Tabii ki bu durumda, ilk serbest bırakma ve son tarih yerine r_i ve d_i mevcut en geç başlama tarihi ve en erken bitiş tarihini belirtmektedir. İşlemin işlem zamanı, işlemin başlangıç ve bitiş zamanı arasındaki farklılık olarak tanımlanmaktadır: proc(A_i) = end(A_i) - start(A_i). p_i, proc(A_i)'nin etki alanındaki en küçük değeri belirtmektedir. Bir işleme ilişkin tüm veriler Şekil 2.2.'de özetlenmektedir. Açık gri, bir işlemin [r_i, d_i] zaman penceresini betimlemek için koyu gri ise, işlemin işletim zamanını belirtmek için kullanılmaktadır[1].



Şekil 2.2. Bir işleme ilişkin veriler [1]

Sonsuz öncelikli çizelgeleme problemleri, çizelgeleme işleminin başlangıç ve bitişinin basit bir kümesinden daha karmaşık olmasından dolayı ifade edilmesi çok daha zordur. Ya her A_i işlemi ile bir set(A_i) değişken kümesi ilişkilendirilebilmekte ya da alternatif olarak her A_i işlemi ve t zamanı için X(A_i, t) 0-1 değişkeni tanımlanabilmektedir. set(A_i), A_i'nin işletildiği zamanlar kümesini temsil etmekte iken X(A_i, t), A_i ancak ve ancak t zamanında işletildiğinde 1 değerini almaktadır. A_i'nin proc(A_i) işletim zamanı, A_i'nin işletildiği t zaman noktalarının sayısı olarak tanımlanmaktadır, örn: | set(A_i) | gibi. Pratikte, X(A_i, t) değişkenleri, t ancak ve ancak set(A_i)'ye ait olduğunda 1 olan X(A_i, t) değeri olarak açık bir şekilde ifade edilememektedir.

Farz edilen zaman ayrıklaştırılmaktadır, start(A_i) ve end(A_i) start(A_i) =min$t \in$ set(A_i) t ve end(A_i) = max $t \in$ set(A_i)t + 1 ile tanımlanabilmektedir. Sonsuz öncelikli olmayan durumda set(A_i)=[start(A_i),end(A_i)) ile [start(A_i),end(A_i)) aralığı soldan kapalı sağdan açıktır yani, | set(A_i) | = end(A_i) - start(A_i) = proc(A_i).

Bu kısıtlar, set(A_i) değişken kümesi için bir üst ve alt sınır sağlanarak kolayca yayılabilmektedir. lb(set(A_i)) alt sınırı, bir seri ayrık ILB_{ui} aralığıdır öyle ki; ILB_{ui}, set(A_i)'ye dahil edilerek kısıtlandırılmaktadır. ub(set(A_i)) üst sınırı, bir seri ayrık IUB_{ui} aralığıdır öyle ki; set(A_i), IUB_{ui} birleşimine dahil edilerek kısıtlandırılmaktadır. Eğer alt sınırın boyutu proc(A_i)'nin üst sınırından daha büyük hale gelirse veya üst sınırın boyutu proc(A_i)'nin alt sınırından daha küçük hale gelirse, bir çelişki algılanmaktadır. Alt sınırın boyutu proc(A_i)'nin üst sınırına (sırayla alt sınır) eşit hale gelirse, set(A_i), son değeri olarak alt sınırı kabul etmektedir. start(A_i) ve end(A_i)'nin minimal ve maksimal değerleri, örn., en erken ve en geç başlangıç ve bitiş zamanları, de korunmaktadır. Biri diğerinden bağımsız olarak düşünülen aşağıdaki kuralların her biri set(A_i), start(A_i) ve end(A_i)'nin sınırlarını güncellemek için kullanılmaktadır. t, zaman içerisinde herhangi bir nokta farz edildiğinde;

$t < r_i \Rightarrow t \notin set(A_i)$

$t \in Ib\big(set(A_i)\big) \Rightarrow start(A_i) \leq t$

$d_i \leq t \Rightarrow t \notin set(A_i)$

$$t \in Ib\big(set(A_i)\big) => t < end(A_i)$$

$$[\forall_{u<t} u \notin ub(set(A_i))] => t \leq start(A_i)$$

$$[\forall_{u\geq t} u \notin ub(set(A_i))] => end(A_i) \leq t$$

$$start(A_i) \leq \max\left\{u \left| \exists_{S\subseteq ub(set(A_i))} |S| = p_i \wedge \min(S) = u \right.\right\}$$

$$end(A_i) \geq \min\left\{u \left| \exists_{S\subseteq ub(set(A_i))} |S| = p_i \wedge \max(S) = u - 1 \right.\right\}$$

Tabii ki, bu kuralların herhangi biri bir değişkenin alt sınırının üst sınırından daha büyük olduğu bir duruma neden olduğunda çelişki algılanmaktadır. Aşağıdaki, kesintiye uğratılamayacak bir A_i işlemi için $X(A_i, t)$ ve $set(A_i)$ notasyonları bazen kullanılabilmektedir. Böyle bir durumda aşağıdaki kurallar da uygulanmaktadır:

$$X(A_i, t) = 0 \wedge t < eet_i => start(A_i) > t$$

$$X(A_i, t) = 0 \wedge lst_i \leq t => end(A_i) \leq t$$

2.3.1.2. Kaynak kısıtları

Bir problemde bulunan kaynak türlerine bakıldığında ayrık çizelgeleme ve kümülatif çizelgeleme ayırt edilmektedir. Ayrık bir çizelgeleme probleminde, kaynaklar 1 kapasitesine sahip olmaktadır ve böylece bir seferde en çok bir işlemi işletebilmektedirler. Kümülatif bir çizelgeleme probleminde, kaynaklar tabii ki kaynak kapasitesinin aşılmamasını sağlayarak paralel çeşitli işlemleri işletebilmektedirler [1].

Kaynak kısıtları şu gerçeği temsil etmektedir ki; işlemler, işletimleri süresince bazı miktarlarda kaynak gerektirmektedir. Bir A_i işlemi ve kapasitesi cap(R) olan bir R kaynağı düşünüldüğünde, genel olarak A_i işlemi tarafından gerektirilen R kaynak miktarını temsil eden bir $cap(A_i,R)$ değişkeni dahil edilmektedir. Hiçbir karışıklık olası olmadığında, çoğunlukla "R" çıkarılmakta ve $cap(A_i,R)$'yi belirtmek için

cap(A_i) kullanılmaktadır. Tam elastik işlemler için cap(A_i,R) değişkeni anlamsızdır ve R kaynağı üzerindeki işlem tarafından gerektirilen enerjiyi temsil eden bir E(A_i,R) değişkeni dahil edilmektedir. Elastik aktiviteler için E(A_i,R)= cap(A_i,R) proc(A_i)'dir. Bir çizelgeyi temsil etmek için bir E(A_i,t,R) değişken kümesi, t zamanında A_i işlemi tarafından kullanılan R kaynak ünitelerinin sayısının belirtilmesini gerektirmektedir. Tüm durumlarda, yeterli kaynak kapasitesinin enerji ihtiyacını kapsamak için işlemlere tahsis edilmesi gerektiğine işaret eden kısıtlar bulunmaktadır:

$$E(A_i, R) = \sum_t E(A_i, t, R)$$

Eğer A_i elastik bir işlem değilse, E(A_i,t,R) ve X(A_i, t) arasında bazı güçlü ilişkiler bulunmaktadır:

$$E(A_i, t, R) = X(A_i, t)\,cap(A_i, R)$$

Elastik işlemler için değişkenler arasında daha zayıf bir ilişki bulunmaktadır:

$$[E(A_i, t, R) > 0] <=> [X(A_i, t > 0)]$$

Genel anlamda, kaynak kısıtı aşağıdaki gibi yazılabilmektedir. T zamanındaki her nokta için

$$\sum_{i=1}^{n} E(A_i, t, R) \leq cap(R) \quad (22.1)$$

Çizelgeleme durumuna bağlı olarak (22.1) tekrar yazılabilmektedir. Sonsuz öncelikli olmayan durumda, (22.1) tüm t zamanları için şuna neden olmaktadır:

$$\sum_{A_i | start(A_i) \leq t < end(A_i)} cap(A_i, R) \leq cap(R)$$

Sonsuz öncelikli durumda, (22.1) tüm t zamanları için şuna neden olmaktadır:

$$\sum_{A_i|start(A_i)\leq t<end(A_i)} X(A_i,t)cap(A_i,R) \leq cap(R)$$

2.3.1.3. Geçici kısıtlar

İşlemler arası geçici ilişkiler, işlemlerin başlangıç ve bitiş değişkenleri arasındaki lineer kısıtlarla ifade edilebilmektedir. Örneğin, A_2'nin başlatılmasının A_1'in bitirilmesinin ardından olacağına işaret eden iki A_1 ve A_2 işlemleri arasındaki standart bir öncelik kısıtı, end (A_1)≤start (A_2) lineer kısıtı ile modellenebilmektedir. Genelde, hem x ve y bir başlangıç ve bitiş değişkeni ve bir d tamsayı, geçici ilişkiler, $x - y \leq d$ türü kısıtla ifade edilebilmektedir.

Temporal kısıt ağı seyrek olduğunda, ki çizelgelemedeki olağan durumdur, bu gibi kısıtlar ayrıt-B-tutarlılık alogaritması kullanılarak kolay yayılabilmektedir.

2.3.1.4. Temel modelin uzantıları

Endüstriyel uygulamalarda sıkça bulunan uzantılardır.

Alternatif kaynaklar

Bazı çizelgeleme durumlarında A_i işlemi, bir S kaynaklar kümesindeki herhangi bir kaynak üzerinde çizelgelenebilmektedir. S'nin A_i için alternatif kaynaklar kümesi olduğu söylenebilmektedir. Bunu modellemenin genel yolu, her A_i işlemi için

kaynak alternatifleri arasından seçilmiş kaynağı temsil eden bir altern(A_i) değişkeni dahil etmektir. Notasyonu basitleştirmek için kaynakların 1'den m'ye kadar sıralandığı ve altern(A_i)'nin değeri üzerinde A_i'nin işletildiği kaynağın indeksini temsil eden değişkeni belirttiği varsayılmaktadır. İşlemin oldukça yaygın işletim süresinin belirli bir işlemin işletildiği kaynağa bağlı olduğuna, örn: kaynakların ilişkili olmaması, dikkat edilmelidir. Aynı durum işlemin işletim maliyeti için de söz konusudur, örn: farklı alternatifler farklı maliyetlere sahip olabilmektedir. Bulunan diğer bir yaygın kısıt türü, kaynak dağılımlarının bağlılığına dayanmaktadır, örn: bir kısıt benzeri "eğer A_1, R_1 kaynağı üzerinde çizelgelenmişse o halde A_3, R_2 kaynağı üzerinde çizelgelendirilmelidir". Bu kısıtlar, alternatif üretim hatlarının modellenmesi için kullanılmaktadır [1].

Alternatif kaynak kısıtları, A_i işlemi her A_{ui} işleminin R_u kaynağı gerektirdiği |domain(altern(A_i))| sözde A_{ui} işlemine ayrılması gibi yayılabilmektedir. Bu notasyonun ardından R_u'i, A_{ui}'nin en erken başlangıç zamanını belirtmektedir vb. Alternatif kaynak kısıtı, $u \in$ domain(altern(A_i)) için A_{ui} alternatif işlemleri arasındaki yapısal ayrıklığı korumaktadır, şunu sağlamaktadır:

$$r_i = \min\{r_i^u | u \in \text{domain}(\text{altern}(A_i))\}$$

$$\text{Ist}_i = \max\{\text{Ist}_i^u | u \in \text{domain}(\text{altern}(A_i))\}$$

$$\text{eet}_i = \min\{\text{eet}_i^u | u \in \text{domain}(\text{altern}(A_i))\}$$

$$d_i = \max\{d_i^u | u \in \text{domain}(\text{altern}(A_i))\}$$

$$\text{Ib}(\text{proc}(A_i)) = \min\{\text{Ib}(\text{proc}(A_i^u)) | u \in \text{domain}(\text{altern}(A_i))\}$$

$$\text{ub}(\text{proc}(A_i)) = \max\{\text{ub}(\text{proc}(A_i^u)) | u \in \text{domain}(\text{altern}(A_i))\}$$

Kısıt yayılımı, alternatif R_u kaynağı üzerinde alternatif A_{ui} işlemleri ortaya çıkarmaktadır. Bir A_{ui} işleminin sınırları anlamsız olduğunda R_u kaynağı, A_i işlemi için olası alternatif kaynaklar kümesinden çıkarılmaktadır, örn: domain(altern(A_i)), domain(altern(A_i)) – {u} haline gelmektedir.

Bazı yaklaşımlarda, sözde A_{ui} işlemleri, orijinal A_i işlemine göre içlerinden birinin alternatif kaynaklardan birini gerçekten gerektireceğini ifade etmek için birlikte oluşturulabilmektedirler. Bu bağlamda, oluşturulan A_{ui} işlemlerden çoğunlukla opsiyonel işlemler olarak bahsedilmektedir.

Kurulum zamanları ve kurulum maliyetleri

Kurulum zamanları ve kurulum maliyetleri endüstriyel uygulamalarda oldukça büyük bir öneme sahiptir. Bol miktarda bulunmaktadırlar ve doğru uygulamaları çoğunlukla oldukça önemlidir, her iki nedenden dolayı gerekli detaydaki problemi ifade etmek için zorunlu bileşenler olmakta iken, gerçek zaman maliyetinin değerli bir parçasını temsil ettikleri için onlara göre iyi bir sonuç bulmak gerekmektedir. A_1 ve A_2 işlemleri arasındaki setup(A_1,A_2) kurulum zamanı, belirli bir kaynak üzerinde A_1 hemen A_2'den önce geldiğinde A_1'in bitiş ve A_2'nin başlangıç zamanı arasında geçmesi gereken zaman miktarı olarak tanımlanmaktadır. Kurulum maliyeti, setupCost(A_1, A_2), A_1 ve A_2 arasındaki geçişle de ilişkili olabilmektedir. Çizelgeleme probleminin amacı, kurulum maliyetleri toplamını minimize eden bir çizelge bulmak olmalıdır. Problemdeki işlemlerin çoğu, aynı makine üzerinde çizelgelenmiş olan kurulumlara bağlıdır (kurulumların anlamsallığı, 1'den daha büyük kapasitedeki kaynaklar üzerinde çok daha fazla karmaşıktır). Kurulum etkenleri, alternatif kaynaklarla birleştirilebilmektedir. Böyle bir durumda, iki parametre her (A_i, A_j, R_u) değişkenler grubu ile ilişkili olmaktadır: A_i ve A_j aynı makinesi üzerinde sıralı olarak çizelgelenmişse A_i ve A_j işlemleri arasındaki setup(A_i, A_j, R_u) kurulum zamanı ve setupCost(A_i, A_j, R_u) kurulum maliyeti. İlgili kısıt şudur ki; start(A_{uj}) ≥ end(A_{ui}) + setup(A_i, A_j, R_u). Ayrıca, A_i, R_u üzerindeki ilk işlem olduğunda A_i'nin başlamasından önce geçmesi gereken setup(-, A_i, R_u) kurulum zamanı ve benzer şekilde A_i, R_u üzerindeki son işlem olduğunda A_i'nin bitişinden sonra geçmesi gereken bir setup(A_i, -, R_u) ayırma zamanı var olabilmektedir [1].

Kırılabilir işlemler ve takvimler

Bir problem, endüstriyel uygulamalarda çokça bulunan pek çok özelliğe sahip olmakla tanımlanmaktadır [24,1]. Böylesi bir özellik, kaynakların, altında işletilen kaynakların çizelgelendiği bir takvim ile yönetilebilmesi gerçeğidir. Ayrıca böylesi bir takvim, işlemler için işletim koşullarını tanımlamakta ve kırılmalar ve üretkenlik profili listesinden oluşmaktadır. Bir kaynak üzerinde elle çizelgelenmiş bir işlem, maksimal kırılma mBD sürekliliğinden daha düşük kırılmalarla kesilebilmektedir. Ayrıca bir işlem mBD>0 olduğunda kırılabilir mBD=0 olduğunda kırılabilir değildir. Üretkenlik profili, işlem işletiminin etkinliğini tanımlamaktadır. Eğer bir işlem % p üretkenlikle bir zaman aralığında çizelgelenmişse, bir işletim zaman ünitesinin % p'si her zaman ünitesi başına işletilmektedir. Ayrıca % 100'ün altındaki bir üretkenlik; bir işletim zaman ünitesinden daha küçük zaman ünitesinin işletileceği anlamına gelmektedir ki o halde bir işlemin sürekliliğinin işletim zamanını aşmayacağını da ima etmektedir. % 100'ü geçen üretkenlikler için açıkça tersi olmaktadır. Bununla birlikte, bir işlemin işletim zamanı üretkenliğin başlangıçtan bitişe kadarki aralığına eşittir.

Bunun için bir KP modeli, her işlem için bir süreklilik değişkeni dahil etmeye ve işletim zaman değişkeninin anlamını tekrar tanımlamaya dayanmaktadır. Bir işlemin dur(A_i) süreklilik değişkeni, işlemin başlangıç ve bitiş zamanı arasındaki farklılık olarak tanımlanmaktadır: dur(A_i) = end(A_i) - start(A_i). İşletim zaman değişkeni, kırılmalarla kesilmediğinde işlemi işletmek için geçecek zaman olarak tanımlanmaktadır ve tüm bu sürede işletim üretkenliği tam anlamıyla % 100'dür. Dört işlem değişkeni, start, end, dur, ve proc, kırılma kısıtı ve işlemin işletildiği kaynağın üretkenlik profili kısıtı ile yönetilebilmektedir [1].

Opsiyonel işlemler/işlemlerin uygulanmadan bırakılması

İster alternatif kaynaklardan gelsin ister modelde direkt olarak bulunsun, çoğu çizelgeleme probleminde işlemler bir kaynak üzerinde neyin işletilip işletilmeyeceğine karar verilmemiş olduğu için var olmaktadırlar. Bu gibi işlemler çoğunlukla opsiyonel işlemler olarak adlandırılmaktadırlar.

Opsiyonel bir işleminin modellenmesi, çoğunlukla işletim zaman değişkeninin 0 değerini alması sağlanarak elde edilmektedir. Standart olmayan öncelik kısıtlarının aktif olmadığına dikkat edilmesi gerekmektedir. Örneğin; opsiyonel A_1 işlemi bir işlemler zincirinin parçası ise ve $end(A_1) + d \leq start(A_2)$ türünde bir öncelik kısıtı tanımlanmakta ise, işletim zaman değişkenini 0 olacak şekilde kurmak ve öncelik kısıtını aktif tutmak, A_1 ve A_2 'nin öncelikleri arasındaki muhtemelen istenmeyen d gecikmesini tetikleyecektir. Opsiyonel işlemleri modellemenin diğer bir yolu, işlemin gerçekten var olup olmadığını belirten her işlem değişkenini dahil etmektir. Bu, modellemenin açıkça daha direkt yoludur fakat, bu ilave değişken ve konseptle ilgilenmek için yayılım algoritmalarının adaptasyonunu gerektirmektedir [25,1].

Endüstriyel çizelgeleme problemlerinde çoğunlukla bir A_i işleminin alt sözleşme olasılığına karşı maruz kalmış belirli bir $cost_i$ maliyeti bulunmaktadır. Bu, işlemin kaynak kapasitesini kullanmadığı fakat zaman aldığı anlamına gelmektedir. Bunu modellemek için bir yol, $cap(A_i)$ kapasite değişkeninin 0 değerini almasına ve $cap(A_i) = 0 \Rightarrow cost = cost_i$ kısıtıyla birlikte maliyeti temsil eden bir maliyet değişkenin dahil edilmesini sağlamaktır. Bu, eğer alt sözleşme, pek tabii yaygın olmayan kurum içi üretimden farklı bir takvimle ilişkili olursa ilginç olabilmektedir. Alternatifin maliyeti, $cost_i$'ye denk gelmektedir.

Durum kaynakları

Bir durum kaynağı, zamanla değişkenlik gösterebilen, sınırsız kapasitenin kaynağını temsil etmektedir. Her işlem, işletim süresi boyunca, belirli bir durumda (veya belirli bir durumlar kümesinde) olan bir durum kaynağı gerektirmektedir. Bundan dolayı, işletimleri süresince bir durum kaynağının uyumsuz durumlarını gerektirirlerse, iki işlem örtüşememektedir. Program kısıtı ve ayrık kısıt adaptasyonları, bu kaynaklar üzerinde temel yayılım algoritmaları olarak kullanılmaktadır.

Depolar

Bir depo kaynağı, işlemlerle tüketilebilen ve/veya üretilebilen çok kapasiteli bir kaynaktır. Bir depo, tamsayı bir maksimal kapasiteye sahiptir ve ilk seviye ye sahip olabilmektedir. Bir depo örneği için bir yakıt tankı düşünülebilmektedir. Kümülatif bir kaynak, işlem kaynağı serbest bıraktığında işlemin başlangıcında tüketilen ve işlemin bitişinde aynı miktarda üretilen özel bir depo örneği olarak görülebilmektedir [1].

Amaç fonksiyonu

Karar problemlerinde, tüm kısıtları karşılayan bir çizelge olup olmadığının belirlenmesi zorunluluktur. Optimizasyon problemlerinde, amaç fonksiyonu optimize edilmek zorundadır.

Bir amaç fonksiyonu modellemenin yaygın yolu, amaç fonksiyonun değerine eşit olması için kısıtlanmış bir kriter kısıtının dahil edilmesidir. Üretim süresinin minimizasyonu, örn: çizelgenin bitiş zamanı, yaygın olarak kullanılmasına rağmen,

diğer kriterler büyük pratik bir öneme sahiptir, örn: kurulum zamanları veya maliyetleri toplamı, gecikmiş işlemlerin sayısı, maksimal veya ortalama yavaşlık veya erkenlik, muhafaza maliyetleri, alternatif maliyetler, yoğun veya ortalama kaynak kullanımı vb.

Klasik çizelgeleme kriterlerinin çoğu, her aktiviteye ulaşılabilen bir δ_i son tarihi hesaba katmaktadır. Zorunlu olan d_i son tarihin tersine, bir δ_i son tarih tercih olarak görülebilmektedir. Aşağıda C_i, A_i işleminin tamamlanma zamanını belirtmektedir. A_i'nin gecikmesi L_i, A_i'nin tamamlanma tarihi ve son tarihi arasındaki fark ile tanımlanmaktadır, örn: $L_i = C_i - \delta_i$. A_i'nin yavaşlığı T_i, max (0, L_i) olarak tanımlanmakta iken A_i'nin erkenliği max (0 - L_i) olarak tanımlanmaktadır. U_i notasyonu, gecikmiş her iş başına bir ünite cezayı belirtmektedir, örn: U_i, $C_i \leq \delta_i$ olduğunda 0'a, tam tersi olduğunda ise 1'e eşittir.

Yaygın olarak çalışılan F kriteri, ya bir toplam ya da bir maksimum olarak formüle edilmektedir. Her işlemin bir w_i ağırlığı, bazı işlemlere daha fazla önem verilmesi için kullanılabilmektedir. Aşağıda iyi bilinen optimizasyon kriterleri bulunmaktadır [1]:

Üretim süresi: $F = Cmax = \max C_i$

Toplam ağırlıklı akış (veya tamamlanma) zamanı: $F = \Sigma w_i C_i$

Maksimum yavaşlık: $F = Tmax = \max T_i$

Toplam ağırlıklı yavaşlık: $F = \Sigma w_i T_i$

Toplam ağırlıklı gecikmiş işlerin sayısı: $F = \Sigma w_i U_i$

Literatürde en çok çalışılmış bu basit durumlar için amaç fonksiyonu, işlemlerin bitiş değişkenlerinin bir fonksiyonudur.

criterion = $F(end(A_1), \ldots, (end(A_n))$

2.3.2. Çizelgeleme problemlerinin modellenmesi

KP gösterilişleri ve tekniklerine dayandırılarak çizelgeleme için belirli bir biçimde çeşitli değişken kısıt tipleri geliştirilmiştir. Değişken tanım bölgeleri şunları içermektedir:

-Her bir değerin bir aralık olduğu aralık etki alanları (örn: başlangıç ve süreklilik)

-Çeşitli kaynak sınıfları için kaynak değişkenleri

-Çizelgeleme uygulamalarında tam sayı değişkenleri zamanlamayı, aralık değişkenleri görevleri, lojik değişkenler karşılıklı bağımlılıkları veya dışlamaları, kaynak etki alanları kaynak sınıflarını belirtmek için kullanılabilmektedir. Yüksek seviyedeki tanım bölgeleri, daha düşük seviyedeki tanım bölgeleri açısından da tanımlanabilmektedir; örneğin, aralık değişkenleri çoğunlukla aralığın başlangıcını ve sürekliliğini belirten tam sayı değişkenleri grubu olarak ifade edilmektedir [3].

Çizelgelemeye özgü kısıtlar şunları içermektedir:

-Aralık değişkenleri için aralık kısıtları (örn: görev 1'in görev 2'den önce meydana gelmesi gerektiğini ifade etmek için $t_1 \leq t_2$)

-Zamanlama değişkenleri için kaynak kısıtları (örn: görevin t aralığı süresince r kaynağında meydana geldiğini ifade etmek için t aralığı ve r kaynağı için ayırma (r,t))

-Ayrıca, daha yüksek seviyeli kısıtlar, daha düşük seviyeli kısıtlar açısından da açıklanabilmektedir. $t_1.s$ başlangıç değişkenleri ve $t_1.d$ (i=1,2; $t_1.d>0$) süreklilik değişkeni ile t_1 ve t_2 aralıkları için $t_1 \leq t_2$ aralık kısıtı, $t_1.s+ t_1.d \leq t_2.s$ tamsayı kısıtına eşittir.

Kaynak kısıtları, bir kaynağın çoklu kullanımlarının nasıl birleştirilebileceğini sınırlayarak kullanılabilir kaynakları tanımlamakta ve kısıtlamaktadır. Kaynaklar, ya yenilenebilir ya da tüketilirdir. r tek terimli kaynaklar bir seferde yalnızca bir görevi

ele almaktadır. Volümetrik kaynaklar (çok birimli veya n'li kaynaklar olarak da adlandırılırlar), görevlerin herhangi bir zamanda kullanılan toplam kaynak miktarının belirli bir kapasite sınırını geçmemesi şartıyla bindirilmesini sağlamaktadır. Durum kaynakları, eğer aynı durumu gerektiriyorlarsa görevlerin bindirilmesini sağlamaktadır. Yenilenebilir kaynakların tersine tüketilir kaynaklar, her kullanımla tükenmektedir ve bariz bir biçimde yenilenmelidir. Son olarak, görevler sonsuz öncelikli ve geçişsiz çizelgeleme arasında farklılığa neden olacak şekilde kesilebilir olabilmekte veya olmayabilmektedir [26,3].

Bir örnek olarak, bir r tek terimli kaynağını iki t_1 ve t_2 görevlerine kısıtları (r, t_1) ayrımı ve (r, t_2) ayrımına aktararak atayabilmekteyiz. Bu kısıtlar, $t_1 \leq t_2 \vee t_2 \leq t_1$ ayırıcı kısıtına eşittir. Kaynak kısıtlarının belirgin kullanımı, çözücünün bu gibi ayırıcı kısıtlarla olduğundan daha fazla etkili temsiller kullanmasını ve çözmesini sağlamaktadır. Çizelgeleme için yararlı örnek bir global kısıt, kümülatif kısıttır: eğer belirli aralıklar t_1, t_1 süresince kaynak kullanımları u_1 ve kaynak kapasitesi k, kümülatif ([t_i,…], [u_i,…], k)'yi karşılarsa, örtüşen görevler ve aynı kaynak bu $\Sigma u \in U \leq k$ zaman noktasındaki U'yu kullanmaktadır. Aralık kısıtları olmayan sistemlerde, bu kısıt kümülatif ([s_i,…], [d_i,…], [u_i,…], k) olarak tanımlanmalıdır k_i, s_i ve d_i sırasıyla aralıkların başlangıç ve süreklilik zamanlarına karşılık gelmektedir [3].

Diğer global kısıt örnekleri, tümden farklı kısıt (farklı değerler almaları için bir kümedeki bütün değişkenleri zorlar) ve nicelik kısıtıdır (belirli bir değerin atanabildiği değişken sayısını sınırlar).

Özgüleme kısıtları ve global arasındaki bir farklılık da şudur ki; global kısıtlar aşamalı olarak tanımlanamamaktadır, örn: tüm görevler kısıt aktarıldığında bilinmelidir ki bu durum onları çevrimiçi çizelgeleme uygulamalarında kullanışsız hale getirmektedir.

Çizelgelemeye özgü yayılma teknikleri

Çoğunlukla kaynaklar ve aralıklar hakkındaki nedenlerle ilgili pek çok çizelgelemeye özgü yayılma teknikleri geliştirilmiştir. Bir teknikte, kaynak programları, herhangi bir zamanda gerekli ve kullanılabilir kapasiteli bir program her bir kaynak için sağlanmıştır. Kaynaklar ve görevler arasındaki yayılma iki şekilde çalışmaktadır: bir görev zamanı ayarlı hale getirilirken görev kaynak kullanımı programa girilmektedir; tam tersine, programda kullanılabilir kapasite düşürülürken ilişkili görevin aralık tanım bölgesi güncellenmektedir.

Bir diğer teknik, kenar algılama, görevlerin belirli bir kaynak üzerinde işletilebildiği sırayı çözmektedir. Her görev, diğer görevler kümesine göre değerlendirilmektedir. Görevin bu görevlerden önce işletilmesi veya işletilemeyeceği belirlenmişse görevin aralık tanım bölgesi üzerinde yeni öncelik kısıtları ve yeni sınırlar çıkarmak olası olabilmektedir.

Çizelgeleme problemi hakkında, yüksek yayılım ve daha küçük arama ağacına neden olan tanım bölgesine özgü kısıtlar ve arama sezgiselleri bulunabilmektedir.

Çevrimiçi çizelgeleme ve model öngörme kontrolü

Model-öngörme kontrolü (MÖK) [27,3], KTÇ'e ilginç sayıda benzerliğe sahip olan popüler bir kontrol yaklaşımı haline gelmiştir. MÖK özellikle, modelin amaçların ve kısıtların bariz bir şekilde kontrol politikasında ayrı bir KOP olarak işaret edildiği model tabanlı bir yaklaşımı ele almaktadır. Aynı zamanda MÖK çizelgelemeyle, işlemin gelen isteklerinin artan doğasını ve bilinen veya tahmin edilen gelecek olayların görüşüne ilişkin kararların optimizsyonunu da paylaşmaktadır. MÖK'ün çeşitli şekillerinin hakkını vermek imkansız olmakta iken KTÇ ve MÖK arasındaki benzerlik ve farklılıkların altını çizmek öğretici olmaktadır [3].

BÖLÜM 3. IBM ILOG CP ÖZELLİKLERİ VE KULLANIMI

IBM ILOG OPL, optimizasyon modellerinin hızlı gelişimi ve yayılımı için mükemmel bir tümleşik gelişim ortamı (TGO) ve modelleme dilidir. Genel amaçlı programlama dillerinden çok daha az çabaya ihtiyaç duyarak herhangi bir optimizasyon modelinin temsili sağlamaktadır. Hata bulucu ve ayar araçları gelişim sürecini desteklemekte ve hazırlanır hazırlanmaz model, bir dış uygulamada uygulanabilmektedir. OPL modelleri, Java, .NET veya C++ ile yazılmış olan herhangi bir uygulamayla kolay bir şekilde birleştirilebilmektedir [28].

3.1. IBM ILOG CP Eniyileme

OPL; çizelgeleme problemlerine özgü sorunları sezgisel ve doğal olarak çözmeye çalışan CP eniyileme motorundaki modelleme özelliklerinden bir çalışma alanı sunmaktadır.

IBM ILOG CP eniyileme, geleneksel matematiksel programlama yöntemleriyle kolay bir şekilde doğrusallaştırılamayan ve çözülemeyen belirgin kombinatoryal optimizasyon problemlerini çözmek için olduğu kadar detaylı çizelgeleme problemlerini de çözümlemek için ikinci kuşak kısıt programlamalardan biridir.

Bir çizelgeleme probleminde en temel işlem, aralıkların başlangıç ve bitiş zamanlarını atamaktır. Çizelgeleme problemleri aynı zamanda, kaynaklar için minimal veya maksimal kapasite kısıtlarının ve bir görevi işleme koymak için alternatif modların yönetimini gerektirmektedir.

Tipik bir çizelgeleme problemi şu şekilde tanımlanmaktadır;

-Zaman aralığı seti – opsiyonel veya zorunlu olabilen işlemlerin, uygulamaların ve görevlerin tanımlamaları

-Zaman kısıtları seti – işlerin, uygulamaların ve görevlerin başlangıç ve bitiş zamanları arasındaki olası ilişkilerin tanımlamaları

-Özel kısıtlar seti – kaynakların durum ve sınırlı kapasitelerinden dolayı aralıklar üzerindeki karmaşık ilişkilerin tanımlamaları

-Bir maliyet fonksiyonu – bazı opsiyonel görevlerin uygulanmama maliyeti veya son tarihi geçen bazı görevlerin teslim edilmesinin ceza maliyeti

Bir çizelgeleme modelinin OPL'deki formatı aşağıdaki gibidir:

-Veri yapısı

-Karar değişkenleri

-Amaç fonksiyonu

-Teknolojik kısıtlar

3.2. OPL ILOG CP Kullanımı

Basit bir konut problemi örneği ile OPL ILOG CP'nin çizelgelemede nasıl kullanıldığı anlatılacaktır. Örnekte belli bir zaman diliminde yapılması gereken görevler listelenmiştir.

Bu görevler, öncelik kısıtlarına sahiptir. Yani; bir görevin başlayabilmesi için başka bir görev tamamlanmalıdır. Örn; örnek kodda çatı yapımı görevinin başlayabilmesinden önce marangozluk görevi tamamlanmalıdır.

using CP;

```
dvar interval duvarorme size 35;
dvar interval marangozluk size 15;
dvar interval sutesisat size 40;
dvar interval tavaninsa size 15;
dvar interval catiyapimi size 5;
dvar interval boyama size 10;
dvar interval pencereler size 5;
dvar interval discehpe size 10;
dvar interval bahce size 5;
subject to {
endBeforeStart(duvarorme, marangozluk);
endBeforeStart(duvarorme, sutesisat);
endBeforeStart(duvarorme, tavaninsa);
endBeforeStart(marangozluk, catiyapimi);
endBeforeStart(tavaninsa, boyama);
endBeforeStart(catiyapimi, pencereler);
endBeforeStart(catiyapimi, discehpe);
endBeforeStart(sutesisat, discehpe);
endBeforeStart(catiyapimi, bahce);
endBeforeStart(sutesisat, bahce);
}
```

OPL'de bir interval karar değişkeni ile temsil edilen zamanın birimi tanımlanmamaktadır. Bunun sonucu olarak bu problemdeki duvar örme görevinin boyutu, 35 saat veya 35 gün veya 35 ay olabilmektedir.

3.2.1. Aralıklar– çizelgelemek için görevler

OPL'de işlemler, aralık tipi karar değişkeni ile temsil edilen aralıklar olarak modellenmektedir. Bir aralık aşağıdaki özelliklere sahiptir:

-Başlangıç
-Bitiş
-Boyut
-Opsiyonellik
-Yoğunluk (takvim fonksiyonu)

Başlangıç ve bitiş arasında kalan zaman, bir aralıktır.

Bir aralığın boyutu, kesinti olmaksızın görevin uygulanması için gereken zamandır. Bir aralık karar değişkeni, bu özelliklerin kısıtlara bağlı olarak modelde çeşitlenmesini sağlamaktadır.

Kodlama Dizimi

dvar interval <gorevadi> <anahtarlar>

<anahtarlar>, aralığa uygulanacak bir veya daha fazla farklı koşulları temsil etmektedir.

Anahtarların bazıları şunları içermektedir;

-Aralık için bir zaman penceresi düzenlemek. Örn; duvar örme işlemi 0. ile 20. günler arasında yapılması için kod şu şekilde olmalıdır.

dvar interval duvarorme in 0..20;

-Aralıklara kendi boyutundan faklı uzunluk koymak. Bir görev değiştirilemez bir boyuta sahip olabilmektedir fakat işlem bir mola süresince askıya alınabilmektedir. Yani, uzunluk boyuttan daha büyük olabilmektedir.

Örneğin; ev örneğindeki pencerelerin kurulması 5 gün sürebilmektedir fakat iş iki gün (hafta sonu) kadar durursa pencerelerin aralık karar değişkeni uzunluğu 7 olmalıdır. Deklarasyon şu şekilde olacaktır:

dvar interval pencereler size 5 in 0..7;

-Opsiyonellik: aralık karar değişkenleri opsiyonel olarak deklare edilebilmektedir. Opsiyonel bir aralık, çözümde bulunabilir de bulunmayabilir de. Peyzaj, ev inşasının gereksiz bir parçası ise aralık karar değişkeni bahçe, opsiyonel olarak deklare edilmelidir;

dvar interval bahce optional;

dvar interval bahce optional in 20..32 size 5;

de şunu deklare etmektedir ki; görev bahce, eğer bulunuyorsa, uygulanmak için 5 zaman birimi gerektirmektedir ve zaman birimi 20'den sonra başlamalı ve zaman birimi 32'den önce bitmelidir. Zaman birimini gün olarak kabul ederek anlatmak gerekirse; bir bahçenin yapımı ev inşası için zorunlu değildir fakat eğer yapılması gerekiyorsa, görev uygulamak için 5 gün gerektirmektedir ve 5 günlük bahçe yapımı, ev inşasının zaman çizelgesindeki 20. ve 32. günleri arasında gerçekleşmelidir.

Aralıklar üzerinde işletilen fonksiyonlar

Aralıklar için uygulanmak üzere birkaç fonksiyon kullanılmaktadır. Bunlar genellikle, bir çizelge için maliyeti tanımlamak veya bir aralığın durumuna erişmek

için karar ifadelerinde (dexpr anahtar sözcüğünü kullanarak) kullanılmaktadırlar. Bunların bazıları aşağıdakileri içermektedir;

-endOf – bir aralığın bitiş zamanına erişmek için kullanılan tamsayı ifadesi

-startOf - bir aralığın başlangıç zamanına erişmek için kullanılan tamsayı ifadesi

-lengthOf - bir aralığın uzunluğuna erişmek için kullanılan tamsayı ifadesi

-sizeOf - bir aralığın boyutuna erişmek için kullanılan tamsayı ifadesi

-presenceOf – opsiyonel bir aralık var ise 1'e, tam tersi ise 0'a dönen tamsayı ifadesi

Yoğunluk (takvim fonksiyonu)

Bir takvim veya yoğunluk fonksiyonu, aralık karar değişkeni ile ilişkilendirilebilmektedir. Yoğunluk, bir aralık uzunluğu üzerinde kullanım veya fayda ölçümü uygulayan bir fonksiyondur. Örneğin; aralık süresince bir kişi veya fiziksel kaynağın mevcudiyetini belirlemek için kullanılabilmektedir.

Kodlama Dizimi

dvar interval < gorevadi > intensity F;

F; aşağıdaki durumlara göre değer alan bir fonksiyondur:

-Yoğunluk, aksi istenmediği takdirde % 100'dür ve bu değer aşılamamaktadır.

-Eğer bir görev belli bir zaman penceresi süresince hiçbir şekilde işleme koyulamayacaksa, bir mola veya tatil gibi, o halde bu zaman periyodu için yoğunluk, 0'a ayarlanmaktadır.

-Bir görev yarı zamanlı işletildiğinde (çalışanın mesai dışı saatleri, diğer görevlerle olan etkileşimler vb. nedeniyle olabilmektedir), yoğunluk pozitif bir yüzde olarak ifade edilmektedir.

Bir haftalık uzunluktaki bir aralık süresince uygulanan bir görev düşünelim (dekorasyon). Bu aralıkta bir çalışan beş tam ve bir yarım gün çalışmakta ve bir gün izin kullanmaktadır; yoğunluk, 5 gün için % 100, bir gün için % 50 ve son gün için sıfır olmalıdır.

Yoğunluk fonksiyonunu, stepFunction OPL anahtar kodu vasıtasıyla lineer aşamalı bir fonksiyon kullanarak deklare edebiliriz.

stepFunction F = stepwise(0—>1; 100–>5; 50–>6; 0–>7);

dvar interval dekorasyon size 5..5 in 1..7 intensity F;

3.2.2. Çizelgeleme kısıtları

OPL ILOG CP eniyileme, çizelgeleme modelini yapılandırmak için özel birtakım kısıtlar sağlamaktadır.

Öncelik kısıtları

Öncelik kısıtları, bir çözümdeki aralık değişkenlerinin göreli pozisyonunu kısıtlamak için kullanılan genel çizelgeleme kısıtlarıdır. Bu kısıtlar, bir aralık değişkeninin diğer bir aralığın başlangıç ve bitiş zamanına göre ne zaman başlayıp bitmesi gerektiğini belirlemek için kullanılmaktadırlar. Sabit veya değişken bir gecikme dahil edilebilmektedir.

Örneğin; bir öncelik kısıtı, bir b işlemi başlamadan önce a işleminin bitmek zorunda olduğu (bazı minimum z gecikmesiyle opsiyonel olarak) gerçeğini modelleyebilmektedir.

OPL'deki öncelikli kısıtların listesi:

-endBeforeStart

-startBeforeEnd

-endAtStart

-endAtEnd

-startAtStart

-startAtEnd

Örnek dizin:

startBeforeEnd (a,b[,z]);

Burada, verilen bir a aralık zamanının sonu (opsiyonel bir zaman değeri z ile modifiye edilmiş), verilen bir b aralık zamanının başlangıcına eşit veya ondan daha küçüktür:

$s(a) + z \leq s(b)$

Ayrıca, boyamanın başlamasından önce tavanlar iki gün süreyle kurutulmak zorundaysa, şu şekilde yazılmalıdır;

endBeforeStart(tavaninsa, boyama, 2);

Kümülatif kısıtlar

Bazı durumlarda, sınırlı kaynakların kullanılabilir olmasından dolayı verilen bir zamanda işleme koyulabilecek aralık sayısı üzerinde kısıtlamalar olabilmektedir. Buna ek olarak, problem tanımlamasında bazı depolar bulunabilmektedir (dolan ve boşalan para akışı veya bir tank).

Kaynak kullanımı üzerindeki bu tip kısıtlar, kümülatif fonksiyon ifadeleri üzerindeki kısıtlarla modellenebilmektedir. Bir kümülatif fonksiyon ifadesi, **cumulFunction** OPL anahtar kodu ile ifade edilmektedir.

Örnek dizin:

cumulFunction <fonksiyonismi> = <temel_fonksiyon_ifadesi>;

<temel_fonksiyon_ifadesi>, modifiye edebilen bir **cumulFunction** ifadesidir. Bu ifadeler aşağıdakileri içermektedir:

-step

-pulse

-tepAtStart

-stepAtEnd

Vurgu fonksiyonu örneği

Vurgu, tek bir aralık değişkeninin veya sabit zaman aralığının kümülatif fonksiyonuna katılımını temsil etmektedir. Bir işlem, kaynak kullanım fonksiyonunu yükselttiğinde veya kaynağı bırakırken kullanımı düşürdüğünde, kümülatif ve yenilenebilir bir kaynağın kullanımını kapsamış olmaktadır.

cumulFunction f = pulse(u, v, h);

cumulFunction f = pulse(a, h);

cumulFunction f = pulse(a, hmin, hmax);

İlgili kodda vurgu fonksiyonu aralığı, herhangi bir a noktası veya başlangıç noktası u ve bitiş noktası v ile ifade edilmektedir. Fonksiyonunu yüksekliği h ile ifade edilmekte veya h_{min} ve h_{max} ile sınırlandırılmaktadır. Açıklamak için, bir kaynağın ne kadarının kullanıldığını ölçen kümülatif bir kaynak kullanımı fonksiyonu düşünelim:

-Zamana bağlı iki aralık bulunmaktadır, A ve B

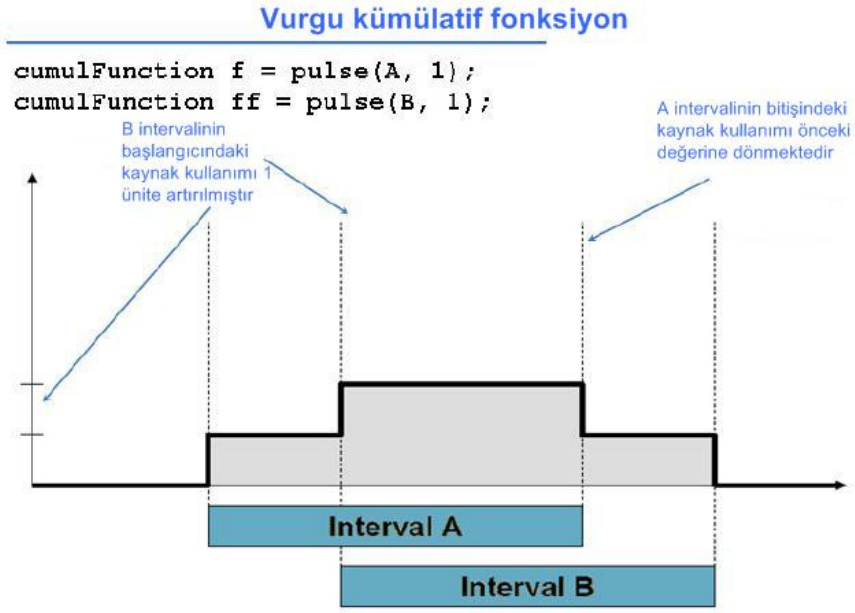
-Her bir aralık, sürekliliği üzerinden bir üniteyle kümülatif fonksiyon ifadesini yükseltmektedir.

Her bir aralık için, kümülatif kaynak kullanımı fonksiyonuna karşılık bu modifikasyon, aralık ve verilen miktarla oluşturulan yalın fonksiyon ile kümülatif fonksiyonun artırılmasıyla yapılabilmektedir.

cumulFunction f = pulse(A, 1);

cumulFunction ff = pulse(B, 1);

Bu göz önüne alındığında fonksiyon, aşağıdaki grafikte gösterilen şekli almalıdır:



Şekil 3.1. Vurgu kümülatif fonksiyon

Basamak fonksiyonu örneği

Basamak, bir noktada başlayan kümülatif fonksiyona karşılık dağılımı ifade etmektedir. Bir noktadan başlayan kullanımı gösteren kümülatif fonksiyondur.

cumulFunction f = step(u, h);

u zamanı, üretim veya tüketimin başlangıcıdır ve h, fonksiyonun yüksekliğini ifade etmektedir.

Başka bir örnek olarak, bütçe kaynağına benzer olarak tüketilebilir bir kaynağı ölçen bir fonksiyon düşünülürse:

-Zaman 2 ye kadar kaynağın kullanımı sıfırdır ve zaman 2 ye geldikten sonra fonksiyon kaynak kullanımını 4 seviye arttırmaktadır.

cumulFunction f = step(2, 4);

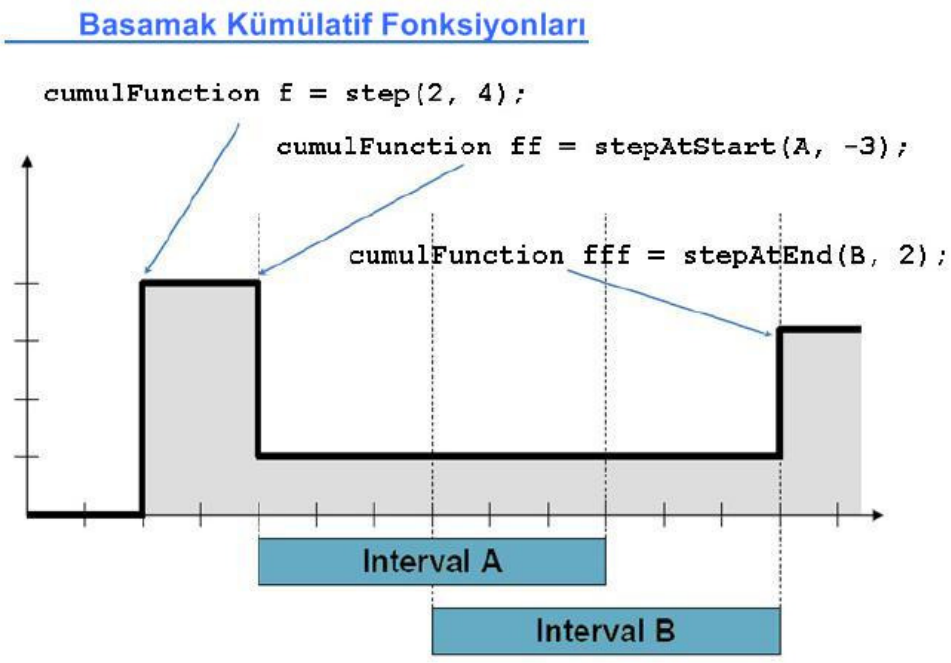
-Zamanla sabit iki aralık bulunmaktadır, A ve B. A aralığı, A aralığı ve 3 değeri ile oluşturulan stepAtStart ın kümülatif fonksiyona uygulanmasıyla modellenen aralığın başlangıcındaki 3 olan kaynak seviyesini düşürmektedir:

cumulFunction ff = stepAtStart(A, -3);

-B aralığı, B aralığı ve 2 değeri ile oluşturulan stepAtStart ın aralık için kümülatif fonksiyona uygulanmasıyla modellenen aralığın bitişindeki 2 olan kaynak seviyesini yükseltmektedir:

cumulFunction fff = stepAtEnd(B, 2);

Bu göz önüne alındığında fonksiyon, aşağıdaki grafikte gösterilen şekli almalıdır:



Şekil 3.2. Basamak kümülatif fonksiyonları

Diğer kümülatif fonksiyon ifadeleri

-stepAtStart – bir aralığın başlangıcından başlayan kümülatif fonksiyona dağılımı ifade etmektedir:

cumulFunction f = stepAtStart(a, h);

cumulFunction f = stepAtStart(a, hmin, hmax);

a aralığının başlangıcı, üretim veya tüketimin başlangıcıdır. Fonksiyonun yüksekliği h ile ifade edilmekte veya hmax ve hmin ile sınırlandırılmaktadır.

-stepAtEnd - bir aralığın bitişinden başlayan kümülatif fonksiyona dağılımı ifade etmektedir:

cumulFunction f = stepAtEnd(a, h);

cumulFunction f = stepAtEnd(a, hmin, hmax);

a aralığının bitişi, üretim veya tüketimin başlangıcıdır. Fonksiyonun yüksekliği h ile ifade edilmekte veya hmax ve hmin ile sınırlandırılmaktadır.

Dizi karar değişkeni ve örtüşmesiz kısıtlar

Bir çizelgeleme modeli, örtüşmemesi zorunlu olan görevler içerebilmektedir, örneğin, belirli bir çalışan tarafından yürütülen görevler aynı anda gerçekleşemezler.

Bunu modellemek için iki yapı kullanılır:

-sequence karar değişkeni

-noOverlap çizelgeleme kısıtı

Öncelikli kısıtlardan farklı olarak, görevlerin göreli pozisyonu üzerinde hiç kısıtlama yoktur. Ek olarak, görevler arasında geçiş süreleri olabilmektedir.

Dizi karar değişkeni

Diziler, dizi karar değişkeni ile ifade edilmektedirler.

Komut Dizini:

dvar sequence <diziismi> in <aralikismi> [types T];

T, negatif olmayan bir tamsayıyı ifade etmektedir.

Bir dizi değişkeni, aralık değişkenleri kümesi üzerinden sıralamayı ifade etmektedir. Eğer bir seq dizesi bir { a1, a2, a3, a4 } aralık değişkenleri kümesi üzerinden tanımlanmakta ise, çözümde bu dize için değer (a1, a4, a2, a3) olabilmektedir. Negatif olmayan bir tamsayı (tip), dizideki her bir aralık değişkeni ile ilişkilendirilebilmektedir. Bu tamsayı, tip değerine göre aralıklar kümesini gruplandırmak için bazı kısıtlar ile kullanılabilmektedir.

Örnek olarak:

dvar sequence calisanlar[w in calisanismi] in

all(h in evler, t in gorevisimleri: calisanlar[t]==w) itvs[h][t] types

all(h in evler, t in gorevisimleri: calisanlar[t]==w) h;

Dizi, aralık değişkenlerinin bir alt kümesini içerebilmekte veya boş olabilmektedir. Bir çözümdeki dizi, çözümde bulunan kümedeki tüm aralıklar üzerinden bir tam sıralamayı ifade edecektir.

Dizideki aralıkların atanan sırası, tam olarak çizelgedeki zamanla göreli pozisyonlarını belirlememektedir. Bir dizenin zamanla göreli pozisyonunu kontrol etmek için aşağıdaki yapılar kullanılır:

-before

-first

-last

-prev

-noOverlap

Örtüşmesiz kısıtlar

Bir dizideki aralıkları kısıtlamak için;

-Dizideki sırayla ilgili olarak zamanla sıralanmaktadırlar

-Örtüşmemektedirler

-Geçiş sürelerine uymaktadırlar

OPL, örtüşmesiz kısıtlar için noOverlap kısıtını kullanmaktadır.

Komut Dizini:

noOverlap (<diziismi> [M]);

<diziismi>, önceden deklare edilmiş bir dizi karar değişkenidir ve M, dizide bir aralığın bitişi ve diğer aralığın başlangıcı arasındaki minimal mesafenin korunması için kullanılabilen opsiyonel bir geçiş matrisidir (çok öğeli küme formunda).

Örnek olarak;

-T[i] tam sayı tipinin n A[i] işlemlerinin bir kümesi, bir makine üzerinde dizilenmektedir.

-t_i tipi işlemi t_j tipi işleme değiştiren abs(ti-tj), dizeye bağımlı kurulum süresi bulunmaktadır.

-Örtüşen hiçbir işlem olmamalıdır.

tuple triplet { int id1; int id2; int value; };

{triplet} M = { <i,j,ftoi(abs(i-j))> | i in Types, j in Types };

dvar interval A[i in 1..n] size d[i];

dvar sequence p in A types T;

subject to {

noOverlap(p, M);

};

noOverlap'ın ilginç bir ek kullanımı, dizenin yararlı olmadığı basit durumlar için aralık dize değişkeninin oluşumunu kısaltmaktır:

noOverlap(A);

şuna eşit olmaktadır:

dvar sequence p in A;

noOverlap(p);

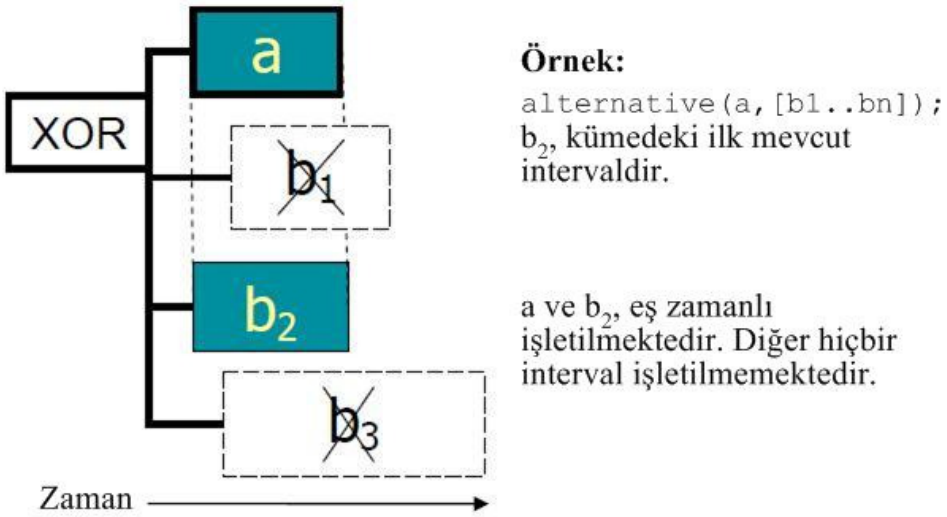
A, bir aralık karar değişkenidir veya bir aralıklar kümesi ve p, bir dize karar değişkenidir.

Alternatif ve yayılma kısıtları

Alternative ve span anahtar sözcükleri, farklı görevlerin işletilmesi ve senkronizasyonunu kontrol etmek için önemli yollar sağlamaktadır.

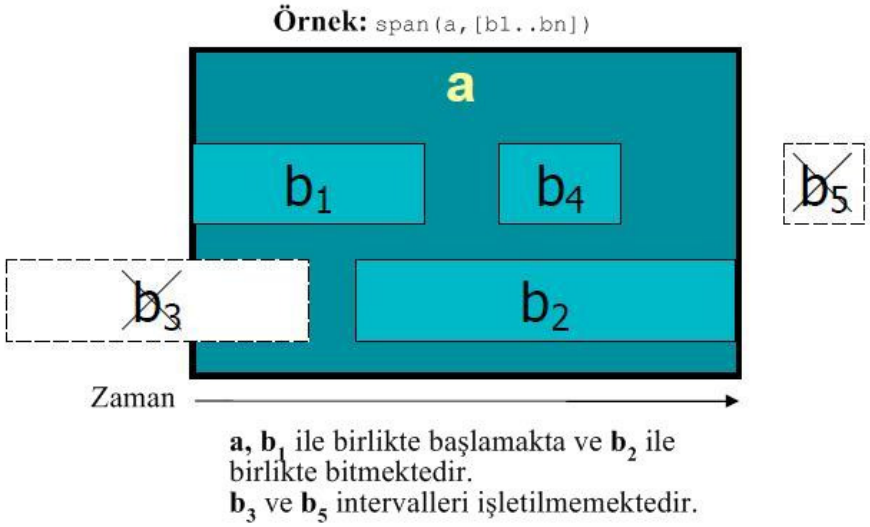
Bir aralık karar değişkeni a ve bir aralık karar değişkenleri kümesi B arasındaki bir alternatif kısıt şuna işaret etmektedir ki; a aralığı, ancak ve ancak B'nin üyelerinin bir tanesinin tam anlamıyla çalıştırılmasıyla çalıştırılmaktadır. Bu durumda, iki görev eş zamanlı hale getirilmektedir.

Yani; a aralığı, B kümesindeki ilk mevcut aralıkla birlikte başlamakta ve onunla birlikte bitmektedir. B kümesinin diğer hiçbir üyesi işletilmemektedir ve a aralığı ancak ve ancak B kümesindeki tüm aralıklar yok olduğunda yoktur.



Şekil 3.3. Alternatif ve yayılma kısıtları örneği

Bir aralık karar değişkeni a ve bir aralık karar değişkenleri kümesi B arasındaki bir yayılma kısıtı şuna işaret etmektedir ki; a aralığı, kümede mevcut tüm aralıkları kapsamaktadır. Yani: a aralığı, B kümesindeki ilk mevcut aralıkla birlikte başlamakta ve B kümesindeki son mevcut aralıkla birlikte bitmektedir. a aralığı ancak ve ancak B kümesindeki tüm aralıklar yok olduğunda yoktur.



Şekil 3.4. Alternatif ve yayılma kısıtları örneği 2

Örnekler:

alternative(tasks[h] [t], all(s in Skills: s.task==t) wtasks[h] [s]);

span(house[i], all(t in tasks : t.house == i) tasks[t]);

Senkronize kısıt

Bir aralık karar değişkeni a ve bir aralık karar değişkenleri kümesi B arasındaki bir senkronizasyon (synchronize anahtar sözcüğü) kısıtı, B kümesindeki tüm mevcut aralıkları a aralığı ile, eğer var ise, aynı anda başlatıp bitirmektedir.

Örnek:

synchronize(task[i], all(o in opers : o.task == i) tiopers[o]);

BÖLÜM 4. DEĞİŞKEN KISITLAMALI GENİŞ ÖLÇEKLİ ENERJİ YÖNETİMİ PROBLEMİNİN KISIT PROGRAMLAMA İLE ÇÖZÜLMESİ

4.1. Problem Tanımı

Fransa'daki EDF (Electricite De France, şirket) enerji üretim tesisleri toplamda 98.8 GW(10^9, GigaWatt) kurulum kapasitesine izin vermektedir. EDF tesislerinin çeşitli enerji formları vardır: termal (nükleer, kömür, fuel oil, gaz), hidrolik ve diğer yenilenebilir enerjiler.

Fransa'da EDF'nin oluşturduğu elektriğin çoğu termal enerji santralleri ile üretilmektedir: 2008 yılında elektriğin %90'ı termal santrallerden üretilmiştir ve bu %90'ın %86'sı nükleer tesislerden elde edilmiştir.

Bu problem, termal enerji santralleri ve özellikle nükleer olanlar gibi yakıt ikmali ve bakım için sürekli olarak kapatılmak zorunda olanlar üzerine odaklanmıştır. Bu hizmet dışı kalmaların programlaması güvenlik, bakım, lojistik ve tesise ilişkin çeşitli kısıtlamalara uygun olmak zorundadır, bütün bu kısıtları sağlarken minimum maliyetli üretim programlarının da yapılması şarttır.

Problem temel anlamda iki alt problemi içermektedir.

-Tesislerin hizmet dışı kalma yani işletim planının belirlenmesi. İşletim planı, yakıt ikmali ve bakım operasyonlarını yerine getirmek için gerekli olan kaynaklar üzerindeki sınırlamalara uygun olmalı yani kısıtlamaları karşılamalıdır.

-Talebi karşılamak için optimal üretim planının belirlenmesi, örn., her bir senaryo için, her bir zaman adımında, her bir tesis tarafından üretilen enerji miktarı.

Amaç**,** beklenen üretim maliyetini minimize etmektir. (Challenge ROADEF/EURO 2010).

4.2. Problem Formülasyonu

Üretim portföyü iki tip üretim ünitesiyle düşünülmektedir:

-Yakıtla devamlı olarak desteklenebilen 1. Tip enerji santralleri

-Düzenli aralıklarla yakıtla desteklenmesi gereken 2. Tip enerji santralleri. 2. Tip enerji santrali ne zaman yeni yakıtla desteklense, kapalı olmak zorunda ve bu hizmet dışı kalma periyodunun uzunluğu boyunca üretim yapamamaktadır.

Bu üretim varlıkları, belirli bir zaman içindeki müşteri talebini karşılamakta kullanılmaktadır. Bu belirli zaman, homojen zaman dilimlerine ayrılmıştır. Müşteri talebi belirsizdir ve talep, belirsizlik senaryolarının uygun bir seti vasıtasıyla bilinmektedir. Bu senaryolar, bazı stokastik süreçlerin gerçekleşmesi olarak varsayılmaktadır.

-1.Tip tesisteki üretim, güç çıktısı ile orantılı olan bir maliyete maruz olmaktadır,

-2.Tip tesislerin maliyeti yüklenen yakıtın miktarı ile orantılı maliyetlere neden olmaktadır. Bu maliyetler aynı zamanda zaman adımına bağlıdır. Zaman tercihinin başlangıcındaki her bir 2. Tip enerji santrali için uygun yakıt miktarı bilinmektedir.

4.2.1. Karar değişkenleri

Aşağıdaki karar değişkenleri problemi oluşturmaktadır:

-Tip enerji santraline yakıt ikmali yapmak için hizmet dışı kalma zaman süreleri

-Tip enerji santraline yakıt ikmali yapılırken sağlanan yakıtın miktarı

-Her bir zaman adımı ve belirsizlik senaryosunda 1. Tip ve 2. Tip enerji santralleri için üretim seviyeleri

4.2.2. Amaç fonksiyonu

Aşağıdaki iki şartın toplamının minimize edilmesi gerekmektedir:

-Tip enerji santrallerinin yakıt ikmalinin maliyeti, tüm senaryolar için belirlenen zamanın sonunda 2. Tip enerji santrallerinde kalan yakıt maliyetten düşmektedir.

-Tüm senaryolar için 1. Tip enerji santrallerinin üretim maliyeti.

Bu amaç, belirlenen tüm zaman adımlarında ve tüm üretim senaryolarında müşteri talebi karşılanıyorken minimize edilmesi gerekmektedir.

4.2.3. Kısıtlar

Her bir senaryo için tüm enerji santrallerinin üretim seviyelerinin toplamının müşteri talebine eşit olması gerekmektedir.

Her bir santral için teknik kısıtlamalar;

-1.Tip enerji santralleri: Üretim seviyeleri MSG (minimum sabit güç) ve tam doluluk arasında olmak zorundadır; tam doluluk seviyesi, senaryo ve zaman adımına bağlı olmaktadır.

-2.Tip enerji santralleri aşağıdaki kısıtlamalara sahiptir:

Üretim seviyeleri MSG ve tam doluluk arasında olmak zorundadır; tam doluluk seviyesi, geçerli zaman adımında mevcut yakıt seviyelerine bağlı olmaktadır. Bu bağımlılık lineer olmayabilir.

Belirlenen zaman süresince yakıt miktarı ve üretim çıktısını ilgilendiren yakıt seviye dinamikleri vardır.

Her bir hizmet dışı kalma boyunca temin edilebilen yakıtın minimum ve maksimum miktarı.

Yakıt ikmali için hizmet dışı kalma zamanında geriye kalan yakıt miktarı üzerine minimum ve maksimum sınır.

Hizmet dışı kalmaların çizelgelenmesi üzerine kısıtlamalar;

-Hizmet dışı kalmalar zaman adımlarının belirli bir miktarı kadar sürmektedir (Ne zaman iki enerji santralinin hizmet dışı kalma süreleri kısmen aynı ana denk gelirse, hizmet dışı kalma sürelerinin örtüştüğünden bahsedilecektir).

-Hizmet dışı kalmalar, zaman adımlarının minimum miktarı ile ayrılmış olmak zorundadır.

-Örtüşen hizmet dışı kalmaların maksimum sayısı

-Spesifik zaman diliminde başlayan hizmet dışı kalmaların maksimum sayısı

4.3. Model Formülasyonu

4.3.1. Modelde kullanılan kelimeler

Zaman adımı: iki ardışık an ile sınırlandırılmış zaman aralığı. Bu problemde zaman aralığı [t, t+1], «*t zaman adımı*» olarak anılmaktadır. Zaman adımları, 1'den başlayarak belirlenen zaman boyunca numaralandırılacaktır.

Hafta: Bir hafta, birkaç zaman adımını temsil etmektedir.

2.Tip enerji santralleri için;

Üretim kampanyası: tesisin üretim yapabildiği süre boyunca zaman adımlarının bir serisine denir.

Hizmet dışı kalma: üretimin mümkün olmadığı süre boyunca zaman adımlarının bir serisine denir. Bir tesise, her hizmet dışı kalmada aşağı yukarı bir kez yakıt ikmali yapılmaktadır.

Bir hizmet dışı kalma yalnızca bir haftanın başlangıcında başlamakta ve bir haftanın sonunda sona ermektedir.

Döngü: bir hizmet dışı kalma ve bir üretim kampanyasının birbirini izlemesi (üretim kampanyası + hizmet dışı kalma).

Yeniden yükleme: bir hizmet dışı kalma süresinde tesise sağlanan enerji miktarı.

Kuplaj: bir üretim kampanyasının ilk haftası.

Dekuplaj: bir hizmet dışı kalmanın ilk haftası.

Modülasyon: bir tesisin maksimum üretkenliği ve güncel üretim arasındaki farklılığa denir.

İndeksler;

s: 0 ve *S-1* arasında değişen senaryolar indeksi

t: 0 ve *T-1* arasında değişen zaman adımları indeksi

h: 0 ve *H-1* arasında değişen haftalar indeksi

j: 0 ve *J-1* arasında değişen 1. Tip enerji santrali indeksi

i: 0 ve *I-1* arasında değişen 2. Tip enerji santrali indeksi

k: 0 ve *K-1* arasında değişen 2. Tip enerji santrali indeksi. *K-1* dikkate alınan periyot süresince başlayabilen döngülerin maksimum sayısını temsil etmektedir.

Veriler;

$DEM^{t,s}$: *s* senaryosunun *t* zaman adımındaki müşteri talebi

D^t: *t* zaman adımının uzunluğu

1.Tip tesisin her j ünitesi için

$PMAX_j^{t,s}$: *s* senaryosunun *t* zaman adımı boyunca maksimum güç

$PMIN_j^{t,s}$: *s* senaryosunun *t* zaman adımı boyunca minimum güç

$C_j^{t,s}$: *s* senaryosunun *t* zaman adımı boyunca orantılı üretim maliyeti

2.Tip tesisin her i ünitesi için:

XI^i: başlangıç yakıtı, örn., ilk zaman adımının başlangıcındaki

$DA_{i,k}$: haftalar içindeki *k* döngüsünün hizmet dışı kalmasının uzunluğu

$MMAX_{i,k}$: *k* döngüsünün üretim kampanyasındaki maksimum modülasyon (bir tesisin maksimum üretkenliği ve güncel üretim arasındaki farklılık)

$RMAX_{i,k}$: *k* döngüsünün hizmet dışı kalması süresince sağlanan maksimum yakıt yükü

$RMIN_{i,k}$: *k* döngüsünün hizmet dışı kalması süresince sağlanan minimum yakıt yükü

$Q_{i,k}$: *k* döngüsünün hizmet dışı kalması süresince yakıt ikmali katsayısı

$PMAX_i^t$: *t* zaman adımı süresince maksimum güç

$BO_{i,k}$: *k* döngüsü süresince güç profili koyma kısıtlamasını aktive eden yakıt stoku üzerindeki sınır

$PB_{i,k}$: *k* döngüsünün üretim kampanyası süresince azalan empoze üretim (parçalı doğrusal konkav işlev, yakıt seviyesine bağımlıdır)

ε: azalan empoze üretimin izlenmesi için tolerans

$SMAX_{i,k}$: *k* döngüsünün üretim kampanyası süresince maksimum yakıt stoku sınırı

$AMAX_{i,k}$: *k* döngüsünün hizmet dışı kalma zamanındaki maksimum yakıt stoku sınırı

Not: bu parametreler şu özelliğe sahiptirler: $SMAX_{i,k} > AMAX_{i,k} > BO_{i,k} > 0$

$C_{i,k}$: *k* döngüsü süresince orantılı yakıt maliyeti

$C_{i,T-1}$: zaman tercihinin sonundaki orantılı yakıt maliyeti [T,T+1]

Kısıtlamalar için;

ha(i,k): *i* tesisinin çevrimdışı gittiği süre boyunca *k* döngüsünün haftası (örn., dekuplaj haftası) h(i,k)=-1 için çizelgeleme olmayacaktır.

p(j,t,s): *s* senaryosunun *t* zaman adımı süresince 1. Tip *j* tesisinin üretimi;

p(i,t,s): *s* senaryosunun *t* zaman adımı süresince 2. Tip *i* tesisinin üretimi;

r(i,k): 2. Tip *i* tesisinin *k* döngüsünün hizmet dışı kalması süresince uygulanan yeniden yükleme;

x(i,t,s): *s* senaryosunun *t* anında (*t* zaman adımının başlangıcındaki) 2. Tip *i* tesisinin yakıt stoku;

ec(i,k): *i* tesisinin *k* döngüsünün üretim kampanyasını oluşturan haftalar seti;

ea(i,k): *i* tesisinin *k* döngüsünün hizmet dışı kalmasını oluşturan haftalar seti;

Tüm kısıtlamalar için 10^{-2} nümerik tolerans varsayılmaktadır.

4.3.2. Kısıtlar

[K1] Talep ve üretimi ilişkilendiren kısıtlama:

Her *s* senaryosunun her *t* zaman adımı boyunca 1. Tip ve 2. Tip enerji santrallerinin toplam üretimi, talebe eşit olmak zorundadır.

$$\forall s, t \sum_{j}^{J} p(j,t,s) + \sum_{i}^{I} p(i,t,s) = DEM^{t,s}$$

Bu, elektrik üretimi ve talebinin her an bir dengede olmak zorunda olduğu gerçeği nedeniyle eşitlik kısıtlamasıdır. Eğer bu denge herhangi bir nedenden dolayı bozulursa, 50 Hz'lik sistem frekansından bir sapma meydana gelmektedir. Böylesi herhangi bir sapma güvenlik sebepleri adına istenmemektedir, sistem frekansını ayar noktasına tekrar geri getirmek için sistem operatörünün aktarımıyla işlemler hemen akabinde gerçekleştirilmek zorundadır.

OPL CP Dilinde Yazımı:

```
forall(t in 1..alltime,s in 1..scenario){
( sum(j in 1..type1) p1[j][t][s] + sum(i in 1..type2) p2[i][t][s] ) == ceil(DEM[s][t] );
}
```

Sum: Toplamı alır

Ceil: Bir üst tam sayıya yuvarlar

1.Tip Enerji Santralleri İle İlgili Kısıtlar

[K2] Üretim üzerindeki sınır:

Her s senaryosunun her t zaman adımı boyunca j tesisinin üretimi minimum ve maksimum sınırlar arasında olmak zorundadır.

$\forall s, t, j\ PMIN_j^{t,s} \leq p(j,t,s) \leq PMAX_j^{t,s}$

OPL CP Dilinde Yazımı:

```
forall(j in 1..type1,t in 1..alltime,s in 1..scenario){
PMIN[j][s][t] <= p1[j][t][s] <= PMAX[j][s][t];
}
```

2.Tip Enerji Santralleri İle İlgili Kısıtlar

[K3] Çevrimdışı üretim:

i tesisinin hizmet dışı olduğu her s senaryosunun her t zaman adımı boyunca üretimi sıfıra eşittir.

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){
( ( t<=endOea[i][k] ) && (t>=strOea[i][k] ) ) =>
p2[i][t][s] == 0;
p2[i][t][s ]<= PMAX2[i][t];
}
```

ea[i][k] // 2.tip tesisin hizmet dışı olduğu t aralığı

strOea[i][k] => strartOf(ea[i][k]) // Hizmet dışı kalmanın başladığı *t* zaman adımı

endOea[i][k] => endOf(ea[i][k]) // Hizmet dışı kalmanın bittiği *t* zaman adımı

[K4] Minimum üretim

i tesisinin hizmet içinde olduğu her *s* senaryosunun her *t* zaman adımı boyunca üretimi pozitiftir veya sıfıra eşittir.

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){
( ( t > endOea[i][k] ) && ( t < strOea[i][k] ) ) =>
0 <= p2[i][t][s];
}
```

[K5] Üretim miktarı kısıtlaması uygulamasının aktivasyonundan önceki maksimum güç

k döngüsünün üretim kampanyasının her *t* zaman adımı ve her *s* senaryosu boyunca, eğer *i* tesisinin mevcut yakıt stoku $BO_{i,k}$'den büyük ya da ona eşitse, üretim seviyesi maksimum sınırına eşit veya ondan daha az olmak zorundadır:

$$\forall s,t,i,k\left(t \in ec(i,k)\right) \wedge (x(i,t,s)) \geq BO_{i,k}) => p(i,t,s) \leq PMAX_i^t$$

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){
( x[i][t][s] >= BO[i][k+1] ) && ( t > endOea[i][k] ) && ( t < strOea[i][k+1] ) =>
modulate[i][t][s]==1;
p2[i][t][s]<=PMAX2[i][t];
}
```

modulate[i][t][s] // üretim kampanyası ve aktivasyon kısıtı sağlanmış ise 1 yoksa 0 dır. Kısıt 12 de kullanılacaktır.

[K6] Üretim miktarı kısıtlaması uygulamasının aktivasyonundan sonraki maksimum güç

k döngüsünün üretim kampanyasının her *t* zaman adımı ve her *s* senaryosu boyunca eğer *i* tesisinin mevcut yakıt stoku $BO_{i,k}$'den aşağı veya ona eşitse, üretim bir ε toleransı ile $PB_{i,k}$ güç profilini izlemek zorundadır:

$\forall s, i, t, k$

eğer $\left(t \in ec(i,k)\right) \wedge \left(x(i,t,s) < BO_{i,k}\right)$

{

eğer $x(i,t,s) \geq (1-\varepsilon)\left(PB_{i,k}\left(x(i,t,s)\right)x(i,t,s)\right) xPMAX_i^t) \times D^t$

ise

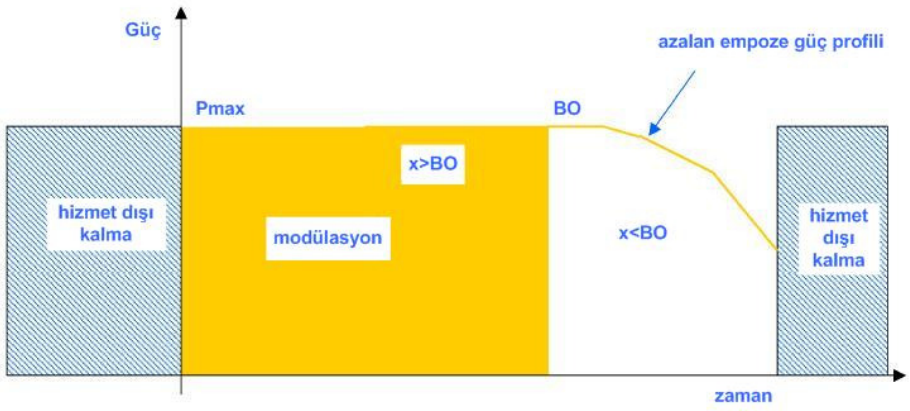
$(1-\varepsilon)\left(PB_{i,k}\left(x(i,t,s)\right)x(i,t,s)\right) xPMAX_i^t) \leq p(i,t,s)$

$\leq (1+\varepsilon)\left(PB_{i,k}\left(x(i,t,s)\right)x(i,t,s)\right) xPMAX_i^t)$

veya

$p(i,t,s) = 0$

}



Şekil 4.1. Azalan empoze güç profili

Bir hizmet dışı kalma; azalan bir güç profili uygulaması öncesinde, süresince veya sonrasında başlayabilmektedir. Eğer üretmek için daha fazla yakıt stoku yoksa üretim sıfıra eşittir ($p(i, t, s)=0$). Üretim, bir hizmet dışı kalma ilan edilmeden önce sınırsız sayıda zaman adımı süresince sıfırda kalabilmektedir (bir hizmet dışı kalmanın en son olası tarihi üzerine olan K13 kısıtlaması sayılmadığı sürece). Bir hizmet dışı kalmayı otomatik olarak başlatan hiçbir olay yoktur.

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign+1){

( x[i][t][s] < BO[i][k] ) && ( t > endOea[i][k-1] ) && ( t < strOea[i][k] ) && ( x[i][t][s] >= ( 1-epsilon ) * pwLinear[i][k][t][s] * PMAX2[i][t] * D[t] ) =>

( 1-epsilon ) * pwLinear[i][k][t][s] * PMAX2[i][t] <= p2[i][t][s];

( x[i][t][s] < BO[i][k] ) && ( t > endOea[i][k-1] ) && ( t < strOea[i][k] ) && ( x[i][t][s] >= ( 1-epsilon ) * pwLinear[i][k][t][s] * PMAX2[i][t] * D[t] ) =>

p2[i][t][s] <= ( 1+epsilon ) * pwLinear[i][k][t][s] * PMAX2[i][t];

( x[i][t][s] < BO[i][k] ) && ( t > endOea[i][k-1] ) && ( t < strOea[i][k] ) && ( x[i][t][s]   < ( 1-epsilon ) * pwLinear[i][k][t][s] * PMAX2[i][t] * D[t] ) =>

 p2[i][t][s] == 0;

 }
```

pwLinear[i][k][t][s] // her *t* zaman adımı ve her *s* senaryosu boyunca *i* tesisinin *k* döngüsünün üretim kampanyası süresince azalan empoze üretimi (bu durum tuple komutu ile sağlanmıştır).

```
tuple powerPr{

key int BreakPoint;

float Slope;
```

}

sorted {powerPr} PB [i in 1..type2][k in 1..campaign+1] = ...;

powerPr PBfirst [i in 1..type2][k in 1..campaign+1] = ...;

dexpr float pwLinear [i in 1..type2] [k in 1..campaign+1] [t in 1..alltime] [s in 1..scenario] = piecewise (o in PB[i][k]) { o.Slope > o.BreakPoint; 0 } (PBfirst[i][k].BreakPoint,PBfirst[i][k].Slope) x[i][t][s];

piecewise // Opl de parçalı doğrusal fonksiyonlar için kullanılır, bu fonksiyon kırılma noktalarına geldiğinde eğimin değeri değişir.

[K7] Yakıt ikmalinde sınırlar

i tesisinin *k* döngüsü süresince gerçekleştirilen tekrar yükleme minimum ve maksimum sınırları içinde olmak zorundadır.

$\forall i, k\ \mathrm{RMIN}_{i,k} \leq r(i,k) \leq \mathrm{RMAX}_{i,k}$

OPL CP Dilinde Yazımı:

forall(i in 1..type2,k in 1..campaign){

RMIN[i][k] <= r[i][k] <= RMAX[i][k];

(lengthOf(ea[i][k]) ==14) || (lengthOf(ea[i][k]) == 7) || (lengthOf(ea[i][k]) == 21)==1;

}

lengthOf(ea[i][k]) // Hizmet dışı kalma aralığının uzunluğu

[K8] Başlangıç yakıt stoku

$\forall s, i \;\; x(i, 1, s) = XI_i$

Başlangıç olduğu için t=1 alınmıştır.

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,s in 1..scenario){
x[i][1][s] == XI[i];
}
```

[K9] Bir üretim kampanyası süresince yakıt varyasyonu

$\forall s, t, i, k \;\; t \in ec(i,k) \Rightarrow x(i,t+1,s) = x(i,t,s) - p(i,t,s) \times D^t$

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){
( t > endOea[i][k] ) && ( t < strOea[i][k+1] )=> x[i][t+1][s]==x[i][t][s]-p2[i][t][s] *
D[t]; }
```

[K10] Bir hizmet dışı kalma süresince yakıt varyasyonu

∀s, t, i, k, (t, ea(i,k)'nın ilk zaman adımıdır)

=> x(i,t+1,s)=(($Q_{i,k}-1$)/$Q_{i,k}$)(x(i,t,s)-$BO_{i,k}$)+r(i,k)+ $BO_{i,k+1}$

Bir 2. Tip enerji santrali yakıt ikmali sürecinde harcanmamış yakıt, yeni yakıt ilavesini olası kılmak için uzaklaştırılmak zorundadır. Yakıt ikmali katsayısı $Q_{i,k}$, bu miktarı belirlemeye yardımcı olmaktadır.

Not: Bir hizmet dışı kalmanın ilk zaman aralığı dışında hiçbir yakıt varyasyonu yoktur.

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){
( t==strOea[i][k] ) => ( x[i][t+1][s] == ceil ( ( ( Q[i][k] - 1.0 ) / Q[i][k] ) * ( x[i][t][s]
- BO[i][k]) + r[i][k] + BO[i][k+1] ) );
( t > strOea[i][k]  &&  t <= endOea [i][k] ) => ( x[i][t+1][s ] == x[i][t][s] );
}
```

[K11] Hizmet dışı kalma anındaki ve yakıt ikmali sonrasındaki yakıt stoku sınırları

∀s, t, i, k, (t, ea(i,k)'nın ilk zaman adımıdır)

$$=> \begin{cases} 0 \leq x(i,t,s) \leq AMAX_{i,k} \\ x(i,t+1,s) \leq SMAX_{i,k} \end{cases}$$

$AMAX_{i,k}$ ve $SMAX_{i,k}$ sınırları özellikle güvenlik nedenlerinden dolayı önemli olmaktadırlar.

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){
( t == strOea[i][k]) => x[i][t][s] <= AMAX[i][k];
( t == strOea[i][k]) => x[i][t+1][s] <= SMAX[i][k];
}
```

[K12] Bir döngü boyunca maksimum modülasyon sınırları

$$\forall s, i, k \sum_{t \in ec(i,k) \wedge x(i,t,s) > BO_{i,k}} [(PMAX_i^t - p(i,t,s)).D^t] \leq MMAX_{i,k}$$

2. Tip enerji santralinin güç çıktısının her modülasyonu, ekipmanın belirli bir miktar aşınmasına neden olmaktadır.

Bununla birlikte, 2. Tip enerji santrallerindeki olağan güç modülasyonları istenmemektedirler.

Not: Eğer $x(i,t,s) \leq BO_{i,k}$ ise maksimum modülasyon kısıtlaması uygulanmamaktadır. Bu durumda kısıtlama K6 güç profilini açıklamaktadır.

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2,s in 1..scenario,k in 1..campaign+1){
sum( t in 1..alltime ) ( modulate[i][t][s] * ( PMAX2[i][t] -p2[i][t][s] ) * D[t] ) <= MMAX[i][k];
}
```

2. Tip enerji santrallerinin hizmet dışı kalmalarının programlanması üzerine kısıtlamalar

[K13] Hizmet dışı kalmanın en erken ve en geç tarihleri

k çevrimde hizmet dışı kalmanın başlangıç zamanı belli bir aralıkta olmak zorundadır

$TO_{i,k}$: hizmet dışı kalmanın başlangıcının ilk olası haftası (örn., dekuplaj)

$TA_{i,k}$: hizmet dışı kalmanın başlangıcının son olası haftası (örn., dekuplaj)

$\forall(\mathrm{i},\mathrm{k})$ $\quad TO_{i,k} \leq ha(i,k) \leq TA_{i,k}$

$\forall(\mathrm{i},\mathrm{k}>0)$ $\quad ha(i,k) \geq ha(i,k-1)+\mathrm{DA}(i,k-1)$

OPL CP Dilinde Yazımı:

```
forall(i in 1..type2, w in east[i]){
```

```
w.week<=hOea[i][w.campaign];
}
forall(i in 1..type2, w in Ista[i]){
hOea[i][w.campaign]<=w.week;
}
forall(i in 1..type2, k in 2..campaign){
hOea[i][k]>=hOea[i][k-1]+DA[i][k-1];
}
```

[K14] Hizmet dışı kalmalar arasındaki minimum aralıklandırma/maksimum örtüşme üzerine kısıtlamalar

Bir A_m setinin hizmet dışı kalmaları, birden M_{14}'e kadar değişen K14 kısıtlamalarının indeksi olan m ile en az Se_m haftaları ile aralıklandırılmak zorundadır.

A_m: nitelendirilir hizmet dışı kalmaların bir seti

Se_m: minimum yetkili aralıklandırma haftalarında süreklilik. Negatif bir değer, maksimum örtüşme olarak yorumlanacaktır.

$\forall(\mathrm{i,k}),(\mathrm{i'},\mathrm{k'}) \in A_m \times A_{m'}(\mathrm{i'},\mathrm{k'}) \neq (\mathrm{i,k}),$

$(\mathrm{ha(i,k)} - \mathrm{ha(i',k')}\text{-}DA_{i',k'} \geq Se_m) \vee (\mathrm{ha(i',k')} - \mathrm{ha(i,k)}\text{-}DA_{i,k} \geq Se_m$

K14'den K18 e kadar olan kısıtlamalar, her bir hizmet dışı kalma süresince gerekli yakıt ikmali ve bakım operasyonlarını uygulamak veya 2. Tip bir enerji santralinin güvenli kuplaj veya dekuplajını sağlamak için mevcut bulunan insan kaynaklarının sınırlı miktarını temsil etmektedirler. K14'ten K20'ye kadar olan kısıtlamalarda, Am

hizmet dışı kalma setleri genel anlamda aynı coğrafik bölgede konuşlanmış olan enerji santrallerinin hizmet dışı kalmalarını temsil etmektedir.

OPL CP Dilinde Yazımı:

```
forall(y in R3,i in A14[y],k in 1..campaign,m in 0..m14){
forall(j in A14[y],s in 1..campaign:j!=i && s!=k){
( hOea[i][k] - hOea[j][s] - DA[j][s] >= Sem14[m] ) + (hOea[j][s] - hOea[i][k] - DA[i][k] >= Sem14[m]) >= 1 ;
}
}
```

[K15] Spesifik zaman periyodu süresince hizmet dışı kalmalar arasındaki minimum aralıklandırma/maksimum örtüşme

[ID, IF] zaman aralığında kesişen Bir A_m setinin hizmet dışı kalması, birden M_{15}'e kadar değişen K15 kısıtlamalarının indeksi olan m ile en az örtüştürülerek veya örtüştürülebilerek aralıklandırılmak zorundadır.

A_m: nitelendirilir hizmet dışı kalmaların bir seti

Se_m: haftalarda minimum yetkili aralıklandırmanın uzunluğu. Negatif bir değer, maksimum yetkili bindirme olarak yorumlanacaktır.

ID_m: spesifik periyodun başlangıç haftası

IF_m: spesifik periyodun son haftası

$\forall(i,k),(i',k') \in A_m \text{ x } A_{m'}(i',k') \neq (i,k)$,

$(ID_m - DA_{i,k} + 1 \leq \text{ha}(i,k) \leq IF_m) \wedge (ID_m - DA_{i',k'} + 1 \leq \text{ha}(i',k') \leq IF_m)$

$\Rightarrow (\text{ha}(i,k) - \text{ha}(i',k') - DA_{i',k'} \geq Se_m) \vee (\text{ha}(i',k') - \text{ha}(i,k) - DA_{i,k} \geq Se_m)$

OPL CP Dilinde Yazımı:

```
forall(y in R2,i in A15[y],k in 1..campaign, m in 0..m15){
forall(j in A15[y],s in 1..campaign:j!=i && s!=k){
( IDm15[m] - DA[i][k] + 1 <= hOea[i][k] <= IFm15[m] ) && ( IDm15[m] - DA[j][s]+1 <= hOea[j][s] <= IFm15[m] ) =>
( hOea[i][k] - hOea[j][s] - DA[j][s] >= Sem15[m] ) + ( hOea[j][s] - hOea[i][k] - DA[i][k] >= Sem15[m] ) >= 1;
}
}
```

[K16] Dekuplaj tarihleri arasında minimum aralıklandırma kısıtlaması

Bir A_m setinin hizmet dışı kalmasının dekuplaj tarihleri, birden M_{16}'e kadar değişen K16 kısıtlamalarının indeksi olan m ile en az Se_m haftaları ile aralıklandırılmak zorundadır.

A_m: nitelendirilir hizmet dışı kalmaların bir seti

Se_m: $Se_m > 0$ ile haftalarda minimum yetkili aralıklandırmanın uzunluğu

$\forall(i,k),(i',k') \in A_m \text{ x } A_{m'}(i',k') \neq (i,k)$,

$| \mathrm{ha}(\mathrm{i},\mathrm{k}) - (\mathrm{ha}(\mathrm{i}',\mathrm{k}')| \geq Se_m$

OPL CP Dilinde Yazımı:

```
forall(i in A16,k in 1..campaign, m in 0..m16){
forall(j in A16,s in 1..campaign:j!=i && s!=k){
abs ( hOea[i][k] - hOea[j][s] ) >= Sem16[m];
}
}
```

[K17] Kuplaj tarihleri arasında minimum aralıklandırma kısıtlaması

Bir A_m setinin hizmet dışı kalmasının kuplaj tarihleri, birden M_{17}'e kadar değişen K17 kısıtlamalarının indeksi olan m ile en az Se_m haftaları ile aralıklandırılmak zorundadır.

A_m: nitelendirilir hizmet dışı kalmaların bir seti

Se_m: $Se_m > 0$ ile minimum yetkili aralıklandırmanın haftalardaki uzunluğu

$\forall(\mathrm{i},\mathrm{k}),(\mathrm{i'},\mathrm{k'}) \in A_m \times A_{m'}(\mathrm{i'},\mathrm{k'}) \neq (\mathrm{i},\mathrm{k}),$

$| \mathrm{ha}(\mathrm{i},\mathrm{k}) + \mathrm{DA}_{\mathrm{i,k}} - (\mathrm{ha}(\mathrm{i}',\mathrm{k}') - DA_{i',k'}| \geq Se_m$

OPL CP Dilinde Yazımı:

```
forall(y in R1,i in A17[y],k in 1..campaign, m in 0..m17){
forall(j in A17[y],s in 1..campaign:j!=i && s!=k){
abs ( hOea[i][k] + DA[i][k] - hOea[j][s] - DA[j][s] ) >= Sem17[m];
}
}
```

[K18] Kuplaj ve dekuplaj tarihleri arasında minimum aralıklandırma kısıtlamaları

Bir Am setinin hizmet dışı kalmasının kuplaj ve dekuplaj tarihleri, birden M18'e kadar değişen K18 kısıtlamalarının indeksi olan m ile en az Sem haftaları ile aralıklandırılmak zorundadır.

Am: nitelendirilir hizmet dışı kalmaların bir seti

Sem: Sem > 0 ile minimum yetkili aralıklandırmanın haftalardaki uzunluğu

$\forall(\mathrm{i},\mathrm{k}),(\mathrm{i'},\mathrm{k'}) \in J \times J,(\mathrm{i'},\mathrm{k'}) \neq (\mathrm{i},\mathrm{k}),$

$|\,\mathrm{ha}(\mathrm{i},\mathrm{k}) + \mathrm{DA}_{\mathrm{i,k}} - \mathrm{ha}(\mathrm{i}',\mathrm{k}')| \geq Se_m$

OPL CP Dilinde Yazımı:

```
forall(y in R,i in A18[y],k in 1..campaign,m in 0..m18){
```

```
forall(j in A18[y],s in 1..campaign:j!=i && s!=k){
abs ( ( hOea[i][k] + DA[i][k] - hOea[j][s] ) ) >= Sem18[m];
}
}
```

[K19] Kaynak kısıtlamaları

Verilen A_m setinin hizmet dışı kalması üzerine kaynakların kullanımı, birden M_{19}'e kadar değişen K19 kısıtlamalarının indeksi olan m ile sınırlı kaynakların sınırlı kullanılabilirliği nedeniyle sınırlamalara bağlı olmaktadır.

A_m: birden M_{19}'e kadar değişen K19 kısıtlamalarının indeksi olan m ile nitelendirilir hizmet dışı kalma seti

$L_{i,k,m}$: belirli bir kaynağın kullanımının başlangıç periyodunu tanımlayan *(i,k)* hizmet dışı kalmanın başlangıcından sonraki haftaların sayısı. L*i,k* , *[0 , DAi,k[* zaman aralığı içinde bulunmaktadır.

$TU_{i,k,m}$: haftalarda kaynak kullanım zamanı ki; bu kaynağın iki ard arda kullanımı arasında olası bulunmazlık periyodu olduğu kadar bu kaynağın olası kullanımının uzunluğunu da içermektedir.

Q_m: kaynağın kullanılabilir miktarı (her hizmet dışı kalma için kullanılan kaynağın miktarı bire eşittir)

Kısıtlama: çözüm ile verilen A_m setinin hizmet dışı kalmaları aşağıdaki koşulları karşılamalıdır:

Q_m'in kullanılabilir miktarı aşılmamıştır kaynağın *$TU_{i,k,m}$* kullanım periyodu dikkate alınmıştır.

$$\forall h \sum_{(i,k)\in A_m} 1_{[ha(i,k)+L_{i,k,m};ha(i,k)+L_{i,k,m}+TU_{i,k,m}[}(h) \leq Q_m$$

Ki $1_A(x)=\begin{cases} 1 \text{ if } x \in A \\ 0 \text{ if } x \notin A \end{cases}$ gösterge fonksiyonudur.

Her durumda,

$L_{i,k,m} + TU_{i,k,m} \leq DA_{i,k,m}$

OPL CP Dilinde Yazımı:

```
forall(m in 0..m19){
sum(k in 1..campaign, h in 1..week,w in Lm[m],n in TUm[m]:w.plant==n.plant)
( ( hOea[w.plant][k] + w.arr_pl[k] <= h && h <= hOea[w.plant][k] + w.arr_pl[k] +
n.arr_pl[k] ) )  <= Qm[m];
}
```

[K20] Verilen bir hafta süresince maksimum örtüşme sayısı üzerine kısıtlama

Verilen bir hafta süresince örtüşme sayısı sınırlandırılmıştır.

Bir *h* haftası için;

Am(h) : hizmet dışı kalmalar seti (*h* ile değişen bir set)

Nm(h): bir *h* haftası süresince paralel durumda hizmet dışı kalmaların maksimum sayısı

Kısıtlama: her *h* için her çözüm adına *Am*(*h*)'nin en çok Nm(*h*) hizmet dışı kalmaları, birden M20'ye kadar değişen K20 kısıtlamalarının indeksi olan m ile *h* haftası süresince birbiri üzerine binebilmektedir.

$$\forall h \sum_{(i,k)\in A_m(h)} 1_{[ha(i,k);ha(i,k)+DA_{i,k}[}(h) \leq N_m\,(h)$$

Ki $1_A(x)=\begin{cases} 1 \text{ if } x \in A \\ 0 \text{ if } x \notin A \end{cases}$ gösterge fonksiyonudur.

OPL CP Dilinde Yazımı:

```
forall(m in 0..m20){
sum ( i in A20,k in 1..campaign) ( ( hOea[i][k] <= Hm[m] ) && ( Hm[m] <= hOea[i][k] + DA[i][k] ) ) <= Nm[m];
}
```

[K21] Bir zaman periyodu süresince enerji santrallerinin bir setinin çevrimdışı güç kapasitesi üzerine kısıtlamalar

Verilen bir periyot için hizmet dışı kalmış olan Cm tesislerinin setinin güç kapasitesi, birden M21'ye kadar değişen K21 kısıtlamalarının indeksi olan m ile maksimum sınırdan aşağıda olmak zorundadır.

Cm: nitelendirilir tesislerin bir seti

ITm: haftalarda bir zaman periyodu

IMAXm: maksimum çevrimdışı güç kapasitesi üzerine sınır

$$\sum_{t\in h\setminus h\in IT_m}\left(\sum_{i\in C_m\setminus(\exists k\setminus t\in ea(i,k))} PMAX_i^t\right)\leq IMAX_m$$

OPL CP Dilinde Yazımı:

forall(m in 0..m21){

sum(i in A21, t in 1..alltime : t>ITm[m][0] && t<=ITm[m][1])PMAX2[i][t] <=IMAX[m];

}

Amaç fonksiyonu

$$\sum_i\sum_k c_{i,k}\,.\,r_i(k)+\frac{1}{|S|}\sum_s\left[\sum_t\left[\sum_j c_j(t,s).p_j(t,s).D^t\right]-\sum_i c_{i,T+1}.x_i\,(T+1,s)\right]$$

T, final zaman adımıdır. $x_i(T+1,s)$, *s* senaryosu üzerindeki son [T,T+1] zaman adımının sonundaki *i* tesisinin arda kalan yakıt miktarını temsil etmektedir.

OPL CP Dilinde Yazımı:

```
minimize sum(i in 1..type2,k in 1..campaign)C2[i][k]*r[i][k] + (1.0/num_scenario) *
sum(j in 1..type1, t in 1..alltime, s in 1..scenario) C1[j][s][t] * p1[j][t][s]*D[t] -sum(i
in 1..type2, s in 1..scenario) C2end[i] * x[i][alltime + 1][s];
```

4.4. Modelin Bütün Kodları

```
/*********************************************

 OPL 6.3 Model

 Author: Erol ASLAN

 *********************************************/

using CP;

int scenario=...;

int type2=...;

int type1=...;

int campaign=...;

int alltime=...;

int week=...;
```

```
int m14=...;

int m15=...;

int m16=...;

int m17=...;

int m18=...;

int m19=...;

int m20=...;

int m21=...;

int zz=...;

int Qm[m in 0..m19]=...;

int Hm[m in 0..m20]=...;

float C2[i in 1..type2][k in 1..campaign]=...;

float C1[j in 1..type1][s in 1..scenario][t in 1..alltime]=...;

float C2end[i in 1..type2]=...;

int num_scenario=card(1..scenario);

int D[t in 1..alltime]=...;

float DEM[s in 1..scenario][t in 1..alltime]=...;
```

```
int BO[i in 1..type2][k in 1..campaign+1]=...;

float PMIN[j in 1..type1][s in 1..scenario][t in 1..alltime]=...;

float PMAX[j in 1..type1][s in 1..scenario][t in 1..alltime]=...;

float PMAX2[i in 1..type2][t in 1..alltime]=...;

int RMIN[i in 1..type2][k in 1..campaign]=...;

int RMAX[i in 1..type2][k in 1..campaign]=...;

float Nm[m in 0..m20]=...;

float IMAX[m in 0..m21]=...;

float epsilon=...;

int DEC_NP=7;//[i in 1..type2][k in 1..campaign];

tuple campaignweek{

int campaign;

int week;

};

{campaignweek} east[i in 1..type2]=...;

{campaignweek} Ista[i in 1..type2]=...;

range R3=0..m14;
```

```
{int} A14[R3]=...;

range R2 =0..m15;

{int} A15[R2]=...;

{int} A16=...;

range R1=0..m17;

{int} A17[R1]=...;

range R =0..m18;

{int} A18[R]=...;

{int} A19=...;

{int} A20=...;

{int} A21=...;

tuple powerPr{

key int BreakPoint;

float Slope;

}

sorted {powerPr} PB [i in 1..type2][k in 1..campaign+1] = ...;

powerPr PBfirst [i in 1..type2][k in 1..campaign+1] = ...;
```

```
int XI[i in 1..type2]=...;

float Q[i in 1..type2][k in 1..campaign]=...;

int AMAX[i in 1..type2][k in 1..campaign]=...;

int SMAX[i in 1..type2][k in 1..campaign]=...;

int MMAX[i in 1..type2][k in 1..campaign+1]=...;

int DA[i in 1..type2][k in 1..campaign]=...;

tuple m19type {

int plant;

int arr_pl[k in 1..campaign];}

{m19type} Lm[m in 0..m19]=...;

{m19type} TUm[m in 0..m19]=...;

int ITm[m in 0..m21][z in 0..zz]=...;

int Sem14[m in 0..m14]=...;

int Sem15[m in 0..m15]=...;

int Sem16[m in 0..m16]=...;

int Sem17[m in 0..m17]=...;

int Sem18[m in 0..m18]=...;
```

```
int IDm15[m in 0..m15]=...;

int IFm15[m in 0..m15]=...;

dvar int+ r[i in 1..type2][k in 1..campaign];

dvar int+ p1[j in 1..type1][t in 1..alltime][s in 1..scenario];

dvar int+ p2[i in 1..type2][t in 1..alltime][s in 1..scenario];

dvar int+ x[i in 1..type2][t in 1..alltime+1][s in 1..scenario];

dvar interval ea[i in 1..type2][k in 0..campaign+1] optional in 0..alltime+1;

dexpr float pwLinear[i in 1..type2][k in 1..campaign+1][t in 1..alltime][s in 1..scenario] = piecewise (o in PB[i][k] ) {o.Slope->o.BreakPoint;0} (PBfirst[i][k].BreakPoint, PBfirst[i][k].Slope ) x[i][t][s];

dvar boolean modulate[i in 1..type2][t in 1..alltime+1][s in 1..scenario];

dexpr int endOea[i in 1..type2][k in 0..campaign+1]=endOf(ea[i][k]);

dexpr int strOea[i in 1..type2][k in 0..campaign+1]=startOf(ea[i][k]);

dexpr float hOea[i in 1..type2][k in 0..campaign+1]=startOf(ea[i][k])/7.0;

minimize sum ( i in 1..type2, k in 1..campaign ) C2[i][k] * r[i][k] + (1.0/num_scenario) * sum ( j in 1..type1, t in 1..alltime, s in 1..scenario ) C1[j][s][t] * p1[j][t][s] * D[t] – sum ( i in 1..type2, s in 1..scenario ) C2end[i] * x[i][alltime+1][s];

subject to{

forall(i in 1..type2){

endOea[i][0]==0;
```

```
strOea[i][0]==0;

endOea[i][campaign+1]==alltime+1;

strOea[i][campaign+1]==alltime+1;

}

//K1

forall(t in 1..alltime,s in 1..scenario){

(sum(j in 1..type1) p1[j][t][s]+ sum(i in 1..type2) p2[i][t][s]) == ceil(DEM[s][t]);}

//K2

forall(j in 1..type1,t in 1..alltime,s in 1..scenario){

PMIN[j][s][t]<=p1[j][t][s]<=PMAX[j][s][t];

}

//K3 ve K4

forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){

((t<=endOea[i][k]) && (t>=strOea[i][k]))=>

p2[i][t][s] ==0;

p2[i][t][s]<=PMAX2[i][t];

}
```

```
//K5

forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){

 (x[i][t][s]>=BO[i][k+1]) && (t>endOea[i][k]) && (t<strOea[i][k+1])=>
modulate[i][t][s]==1;

}
//K6

forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign+1){

(x[i][t][s]<BO[i][k]) && (t>endOea[i][k-1]) && (t<strOea[i][k]) &&
(x[i][t][s]>=pwLinear[i][k][t][s]*PMAX2[i][t]*D[t])=>

  (1-epsilon)*pwLinear[i][k][t][s]*PMAX2[i][t] <= p2[i][t][s];

(x[i][t][s]<BO[i][k]) && (t>endOea[i][k-1]) && (t<strOea[i][k]) &&
(x[i][t][s]>=pwLinear[i][k][t][s]*PMAX2[i][t]*D[t])=>

p2[i][t][s] <= (1+epsilon)*pwLinear[i][k][t][s]*PMAX2[i][t];

(x[i][t][s]<BO[i][k]) && (t>endOea[i][k-1]) && (t<strOea[i][k]) &&
(x[i][t][s]<pwLinear[i][k][t][s]*PMAX2[i][t]*D[t])=>

p2[i][t][s]==0;

 }

//K7

forall(i in 1..type2,k in 1..campaign){

RMIN[i][k]<=r[i][k]<=RMAX[i][k];

(lengthOf(ea[i][k])==14)||(lengthOf(ea[i][k])==7)||(lengthOf(ea[i][k])==21)==1;
```

```
}

//K8

forall(i in 1..type2,s in 1..scenario){

x[i][1][s]==XI[i];
}

//K9

forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){

 (t>endOea[i][k]) && (t<strOea[i][k+1])=> x[i][t+1][s]==x[i][t][s]-p2[i][t][s]*D[t];

 }

//K10

forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){

  ( t==strOea[i][k] ) => ( x[i][t + 1][s] == ceil( ( ( Q[i][k]-1.0) / Q[i][k] ) * ( x[i][t][s]
- BO[i][k] ) + r[i][k] + BO[i][k + 1] ) );

 ( t>strOea[i][k] && t<=endOea[i][k])=> (x[i][t+1][s]==x[i][t][s]);

 }

 //K11

 forall(i in 1..type2,t in 1..alltime,s in 1..scenario,k in 1..campaign){

 ( t==strOea[i][k]) =>x[i][t][s]<=AMAX[i][k];

 ( t==strOea[i][k]) => x[i][t+1][s]<=SMAX[i][k];
```

```
}

//K12

forall(i in 1..type2,s in 1..scenario,k in 1..campaign+1){
sum (t in 1..alltime) ( modulate[i][t][s] * (PMAX2[i][t] - p2[i][t][s]) * D[t]) <= MMAX[i][k];

}

 //K13

forall(i in 1..type2, w in east[i]){

w.week<=hOea[i][w.campaign];

 }

forall(i in 1..type2, w in Ista[i]){

hOea[i][w.campaign]<=w.week;

}

forall(i in 1..type2, k in 2..campaign){

hOea[i][k]>=hOea[i][k-1]+DA[i][k-1];

}

 //K14

forall(y in R3,i in A14[y],k in 1..campaign,m in 0..m14){

forall(j in A14[y],s in 1..campaign:j!=i && s!=k){
```

```
( hOea[i][k]- hOea[j][s] - DA[j][s] >= Sem14[m]) + (hOea[j][s]-hOea[i][k]-DA[i][k]
>= Sem14[m]) >= 1 ;


}
}

forall(y in R3,i in A14[y],k in 1..campaign,m in 0..m14){

forall(y in R3,j in A14[y],s in 1..campaign:j!=i && s!=k){

( hOea[i][k] - hOea[j][s] - DA[j][s] >= Sem14[m] ) + ( hOea[j][s] - hOea[i][k] -
DA[i][k] >= Sem14[m]) >= 1 ;


}

}

//K15

forall(y in R2,i in A15[y],k in 1..campaign, m in 0..m15){

forall(j in A15[y],s in 1..campaign:j!=i && s!=k){

( IDm15[m] - DA[i][k] + 1 <= hOea[i][k] <= IFm15[m] ) && ( IDm15[m] -DA[j][s]
+ 1 <= hOea[j][s] <= IFm15[m] ) =>

( hOea[i][k] - hOea[j][s] - DA[j][s] >= Sem15[m] ) + ( hOea[j][s] - hOea[i][k] -
DA[i][k] >= Sem15[m] ) >= 1;


}

}

//K16

forall(i in A16,k in 1..campaign, m in 0..m16){
```

```
forall(j in A16,s in 1..campaign:j!=i && s!=k){
abs(hOea[i][k]-hOea[j][s])>=Sem16[m];
}
}
//K17
forall(y in R1,i in A17[y],k in 1..campaign, m in 0..m17){
forall(j in A17[y],s in 1..campaign:j!=i && s!=k){
abs(hOea[i][k]+DA[i][k]-hOea[j][s]-DA[j][s])>=Sem17[m];
}
}
//K18
forall(y in R,i in A18[y],k in 1..campaign,m in 0..m18){
forall(j in A18[y],s in 1..campaign:j!=i && s!=k){
abs((hOea[i][k]+DA[i][k]-hOea[j][s]))>=Sem18[m];
}
}
//K19
```

```
forall(m in 0..m19){

sum(k in 1..campaign, h in 1..week,w in Lm[m],n in TUm[m]:w.plant==n.plant)

(( hOea[w.plant][k] + w.arr_pl[k] <= h && h <= hOea[w.plant][k] + w.arr_pl[k] +
n.arr_pl[k] ) )

<= Qm[m];

}

 //K20

forall(m in 0..m20){

sum(i in A20, k in 1..campaign) ( ( hOea[i][k] <= Hm[m] ) && (Hm[m] <=
hOea[i][k] + DA[i][k] ) ) <= Nm[m];

}

//K21

forall( m in 0..m21){

sum(i in A21, t in 1..alltime:t > ITm[m][0] && t <= ITm[m][1]) PMAX2[i][t] <=
IMAX[m];

}

};
```

4.5. Veri Yapısı

Uygulama için EK A verisi kullanılmıştır. Veri yapısı genel olarak şu özelliklere sahiptir.
-Senaryo sayısı 2'dir.

-Kampanya sayısı 2'dir.

-1.Tip tesis sayısı 1'dir.

-2.Tip tesis sayısı 2'dir.

-Sağlanması istenilen kısıtlar; K1, K2, K3, K4, K5, K6, K7, K8, K9, K10, K11, K12, K13 (4 tane), K14 dür.

Verinin ham ve düzeltilmiş şekli ekte yer almaktadır.

Problemin çözümü için kullanılacak verinin formatı .txt'dir fakat IBM ILOG CP nin direkt kullanacağı bir yapıda değildir.

Örneğin 2. Tip tesisten 2 adet olduğunu düşünürsek ve kampanya sayısı da 2 ise bu tesislerin AMAX ve SMAX ham veri yapısı aşağıdaki gibidir.

type 2

index 0

max_stock_before_refueling 3175200 3175200

max_stock_after_refueling 14112000 14112000

type 2

index 1

max_stock_before_refueling 3304800 3304800

max_stock_after_refueling 14688000 14688000

Bu ham veri yapısı OPL'e uygun hale getirilmek için aşağıdaki gibi düzeltilmelidir.

```
AMAX=[[3175200,3175200]
[3304800,3304800]
];

SMAX=[[14112000,14112000]
[14688000,14688000]
];
```

Veri yapısının düzeltilmiş hale getirilmesi için Visual.Net kullanılarak bir yazılım oluşturulmuştur. Yazılımın kodları EK B de sunulmuştur.

4.6. Visual.Net İle Model İçin Arayüz Oluşturulması

OPL ILOG ile oluşturulan model için Visual Studio ortamında ara yüz oluşturulmuştur. Bu arayüz oluşturulurken aşağıdaki durumlar göz önüne alınmıştır.

-Model tanımlamasının belirlenmesi

-Bir veri kaynağının tanımlanması

-Çözüm içindeki karar değişkenlerine erişimin sağlanması

-Model öğelerine erişimin sağlanması

Arayüz kodları Vb.Net ile aşağıdaki gibi yazılmıştır;

```
Imports ILOG.OPL

Imports ILOG.Concert

Imports ILOG.CP

Module Mulprod

  Const DATADIR As String = "../.."

  Sub Main()

    Dim status As Integer = 127

    Try

      OplFactory.DebugMode = True
      Dim oplF As OplFactory = New OplFactory

      Dim errorHandler As OplErrorHandler = oplF.CreateOplErrorHandler()

          Dim modelSource As OplModelSource =
oplF.CreateOplModelSource(DATADIR + "/roadef.mod")
      Dim settings As OplSettings = oplF.CreateOplSettings(errorHandler)

      Dim def As OplModelDefinition =
oplF.CreateOplModelDefinition(modelSource, settings)
```

```
        Dim cp As CP = oplF.CreateCP()

        cp.SetOut(Nothing)

        Dim opl As OplModel = oplF.CreateOplModel(def, cp)

        Dim dataSource As OplDataSource = oplF.CreateOplDataSource(DATADIR
+ "/data0.dat")

    opl.AddDataSource(dataSource)

    opl.Generate()

        If (cp.Solve()) Then

            Console.Out.WriteLine("OBJECTIVE: " + Str(opl.CP.ObjValue))

            opl.PostProcess()

            opl.PrintSolution(Console.Out)

            status = 0

        Else
            Console.Out.WriteLine("No solution!")
            status = 1

        End If
```

```
      oplF.End()

    Catch ex As ILOG.Concert.Exception

      Console.WriteLine(ex.Message)

      status = 2

    End Try

    Environment.ExitCode = status

    Console.WriteLine("--Press <Enter> to exit--")

    Console.ReadLine()
  End Sub
End Module
```

4.7. Modelin Çözümü ve Sonuçlar

Model çözülmüş ve şu sonuçlar elde edilmiştir. Toplam maliyet 8738522417232.56 çıkmıştır. Örneğin yüklenecek yakıt miktarı 2.tip tesisin 1.senaryosunun 1. kampanyası için 9102240 çıkmıştır. 1.tip tesisin 1.senaryo için 1. t zamanında üreteceği enerji miktarı 48649 dür. 1.tip tesisin 1. t zamanında kullandığı yakıt miktarı 4974480 dür. 2.tip tesisin 1. senaryosunun 1. t zamanında üreteceği enerji miktarı 1335 dir. Bütün sonuçlar aşağıda verilmiştir.

```
// amaç fonksiyonun çözümü 8738522417232.56
```

r = [[9102240 9102240] [6462720 6462720]];

p1 = [[[48649 48960] [47177 45956] [49169 49904] [49285 49660] [48510 50465] [47878 50394] [48373 50276] [48632 50519] [46885 47311] [52059 50313] [52105 50725] [52832 50769] [52043 51150] [52807 52252] [49848 49754] [47489 45990] [50886 51359] [52391 50472] [52746 49347] [52588 49205] [52882 49673] [52626 47665] [50473 45170] [53316 52164] [54180 53043] [56739 53457] [56120 51925] [55833 51846] [52755 48399] [50562 46851] [55888 51385] [55659 52455] [57456 52345] [55483 51543] [56305 52355] [53542 52498] [51043 50206] [56682 53820] [55608 53883] [56968 55188] [58315 56043] [59588 56197] [57572 51516] [60086 52522] [58809 57466] [60067 57305] [59838 54724] [60936 55506] [59967 56699] [58795 57536] [56430 55734] [59911 59225] [59376 59683] [62864 62948] [63257 64132] [65773 64658] [60246 60927] [56407 55308] [62504 61744] [61853 62430] [60816 64119] [60305 61477] [54041 51349] [58478 51658] [56314 51329] [61667 61820] [63026 62035] [62038 61774] [61424 59359] [60838 58191] [57807 54952] [56776 54262] [64937 60917] [65622 59123] [68346 59867] [66949 64152] [69160 62273] [66130 58643] [64093 57896] [70023 66651] [70298 67586] [70200 66344] [69770 71681] [72884 72528] [65946 70999] [64192 67052] [71574 68069] [73179 70091] [73743 67699] [71013 70159] [71631 70998] [67266 62246] [64039 58594] [69454 66452] [73545 69091] [72838 67169] [71295 65974] [73890 67482] [67042 63009] [63034 59748] [66976 64634] [63869 66994] [66545 67868] [71308 67852] [71665 65698] [67015 59244] [65127 55749] [69427 64239] [69589 67237] [68366 70403] [68946 72654] [66898 72421] [61575 67446] [57400 64913] [61760 75291] [67528 78898] [69018 80505] [66926 78129] [66208 80087] [61076 76683] [60511 74065] [70247 79933] [71449 77625] [72481 79706] [71368 77669] [74542 79811] [74131 75322] [72450 75063] [79816 76765] [80405 77544] [79528 77819] [77768 73686] [77473 70802] [69504 63723] [66239 61525] [70216 68929] [69172 70865] [69773 70454] [67992 68320] [67175 69258] [61322 64472] [58886 63268] [67205 70473] [66230 70783] [64703 68792] [63145 70781] [63272 73744] [57307 69484] [52173 68925] [58235 69935] [59704 70983] [61421 67962] [61240 68482] [61633 67000] [55859 60129] [54055 58079] [60462 64528] [58423 66979] [58214 67172] [58218 66513] [58225 64242] [55447 53510] [53566 53973] [56101 63837] [56416 65731] [56528 63210] [56717 62630] [57843 64728] [56496 62772] [54566 60634] [57135 66336] [57235 62387] [57274 62228] [56586 62258] [56947 62743] [53072 57320] [53299 55002] [56348 60202] [55141 59040] [54516 59570] [54903 60169] [55580 57292] [54673 57541] [53224 55282] [57113 59414] [54263 58732] [56163 59300] [54516 56732] [54881 61233] [50407 57989] [50614 57523] [55372 57061] [54337 57129] [57485 59835] [54812 58600] [54009 57864] [54188 58967] [50998 54762] [57229 58004] [56606 58220] [56300 57963] [56918 62253] [57655 60936] [50614 60682] [50211 59350] [52422 58410] [51377 60883] [51926 58729] [53909 59840] [53455 59824] [50805 57484] [48985 56421] [53527 57748] [54137 58501] [54162 57498] [53948 55674] [53676 55852] [47366 53193] [45779 50740] [46181 50631] [52324 53732] [54413 55988] [56157 56240] [53050 56455] [51960 53304] [49170 50295] [54001 55625] [54971 56627] [55318 55527] [53796 55940] [55579 55766] [48443 51953] [42647 47696] [43126 48257] [50537 55436] [53563 57008] [53610 55124] [53218 57440] [49145 53430] [46403 51898] [47959 52676] [51213 54290] [52444 52545] [49156 52453] [50177 54773] [48066 52663] [45664 49589] [50285 53912] [51890 52878] [51533 53848] [53161 54629] [51390 54354] [47602

51545] [44910 48190] [49977 53166] [49650 53822] [47519 52645] [41996 47342] [44584 47110] [44026 47121] [43722 45287] [50827 52123] [51399 52004] [50411 51879] [48201 51567] [46247 50575] [42649 47245] [39564 43318] [46253 49635] [46553 48002] [48488 47830] [48927 50387] [48463 50318] [42265 46822] [41231 43847] [46871 49818] [44594 49474] [45607 51384] [46860 51305] [46776 51192] [46382 49222] [44128 46825] [50638 51915] [50277 52323] [50899 52554] [47940 51326] [50980 53898] [45317 50035] [43173 48053] [48078 51191] [47992 50715] [48991 52097] [48142 50606] [48719 50194] [48561 48440] [45241 45936] [50492 50998] [49704 51100] [50004 49921] [50459 49985] [51744 51183] [45200 49948] [45397 48149] [47405 52296] [48983 50939] [49251 50377] [48519 51308] [46284 45771] [48681 48892] [46359 46593] [49597 52338] [50878 52892] [51311 53809] [50593 54384] [50052 54015] [49350 50442] [47207 47518] [52322 53659] [53437 55188] [53927 55543] [53896 55199] [53404 54372] [49233 50425] [45701 46233] [50054 49929] [49226 49614] [50787 48725] [49860 48164] [50056 47793] [46864 44907] [43953 41408] [47952 44663] [48696 43238] [48508 45051] [48679 43411] [49316 46379] [46944 45243] [43878 41874] [48390 44839] [46143 43400] [50327 47009] [50727 47607] [51081 48491] [47575 46057] [44884 42641] [49438 49236] [51706 49280] [52128 49369] [52366 49991] [53328 50111] [49922 47465] [46583 44469] [51697 51246] [51863 52653] [53939 52116] [52895 50620] [55160 52330] [52858 49184] [50037 47604] [53051 52610] [54406 52387] [54615 52615] [54814 54028] [52626 52537] [49588 49224] [47958 47007] [54334 52082] [54648 52786] [50633 52694] [50543 54894] [52669 53686] [48703 42677] [46189 40340] [50489 48229] [49290 50100] [50465 51836] [50989 51298] [51750 51408] [48507 46601] [47019 45276] [50298 51087] [52338 51824] [53543 52172] [52078 51828] [52706 49568] [51243 48053] [49519 50502] [54885 56285] [56162 53844] [53657 56289] [53711 54520] [55426 58292] [51211 54687] [48921 52565] [52775 57527] [53288 58445] [56127 58753] [57366 59282] [57513 60174] [51512 56523] [49081 51759] [54337 57803] [56790 57854] [58180 57785] [55458 57947] [56401 57972] [54105 54179] [53829 52733] [58716 60285] [58890 59513] [55668 60387] [58668 62231] [57851 62061] [54104 61306] [46226 60127] [53667 62463] [55936 65465] [57873 63545] [59691 67869] [60607 67825] [59169 67365] [56068 64151] [60801 68970] [61027 68155] [62040 67001] [60466 67550] [60512 67544] [56712 56166] [57385 55868] [62524 65927] [60243 67739] [59535 68621] [59689 69275] [58783 66734] [57837 61883] [57007 59738] [61451 66803] [60234 67017] [63719 66512] [60921 66423] [62694 66851] [67845 62304] [64737 60888] [67219 66959] [70752 69884] [69043 70709] [68464 70180] [69066 70324] [64416 64149] [62119 65189] [69662 71863] [72465 71256] [73355 70749] [76275 67918] [74890 69675] [70629 62516] [67568 56868] [74369 64497] [72873 64061] [72386 64946] [72167 66821] [73008 67608] [72247 63414] [66742 60984] [73257 69857] [71541 66853] [71077 68561] [70366 68977] [73052 69121] [70865 66429] [69009 60131] [70141 59194] [75767 67265] [78084 69702] [78354 72512] [75401 74763] [70049 72272] [66710 67380] [64034 69633] [68915 78403] [73126 81619] [74136 83166] [71270 83171] [66628 79498] [65446 78730] [73400 81270] [75550 82345] [76069 82165] [76355 81230] [77025 79712] [77635 80935] [77031 77907] [80360 80694] [78291 80872] [78110 82427] [77851 83605] [76329 79489] [72200 71166] [70505 67285] [74382 73382] [72879 74301] [73286 70447] [74226 71280] [73200 70308] [65009 66022] [62728 64621] [68222 73978] [70262 74262] [72148 73586] [68746 77061] [68366 78424] [61458 74603] [59645 73309] [64981 76040] [65939 77297] [65454 74169] [66693 73761] [67000 73999] [63917 66695] [62379 63320] [68556 71171] [67780 70806]

[69426 72053] [67224 73076] [67449 71419] [58637 64614] [53786 61865] [62324 66380] [64534 66690] [63711 68066] [62873 69742] [64745 72117] [58013 67052] [54854 65241] [59328 67901] [57763 65279] [58216 67509] [59408 65604] [64311 64395] [63263 57394] [59812 55723] [65437 63768] [61513 62753] [61174 60474] [63296 62952] [62765 65060] [61128 60947] [60739 57329] [60489 61562] [61217 61714] [62404 62707] [60946 62103] [61805 64194] [60922 60936] [60824 58239] [67368 62048] [65723 61714] [64505 63516] [64082 63270] [67821 64986] [58954 59558] [56017 56854] [58238 59015] [59077 57042] [60725 57127] [61804 57558] [60881 58806] [57696 63387] [53584 58689] [55968 62991] [52161 59481] [55002 61092] [53407 60771] [56550 60638] [52229 56744] [50446 55109] [51664 56461] [54113 57406] [54565 60696] [52797 57165] [52809 58158] [52065 55876] [49424 48777] [53436 55020] [56182 53471] [55080 55219] [55511 52574] [55080 53104] [53491 50550] [50803 47489] [56376 51894] [57455 53605] [57478 56753] [57673 56678] [60419 54587] [53104 54553] [48576 49663] [50179 50193] [43842 49323] [49534 54828] [52702 54778] [51101 55634] [48606 53915] [46212 50166] [52735 55102] [49247 53891] [54396 56961] [54848 57060] [53621 57752] [47580 52750] [41979 49448] [50294 55770] [52435 56408] [53428 55365] [43874 49040] [47797 49189] [46019 49641] [40200 45775] [48771 52940] [49112 54933] [49577 55289] [49832 55263] [48509 54314]]];

x = [[[4974480 4974480] [4942440 4942440] [4910400 4910400] [4878360 4878360] [4846320 4846320] [4814280 4814280] [4782240 4782240] [4750200 4750200] [4718520 4718520] [4686840 4686840] [4655160 4655160] [4623480 4623480] [4591800 4591800] [4560120 4560120] [4528440 4528440] [4496760 4496760] [4465080 4465080] [4433400 4433400] [4401720 4401720] [4370040 4370040] [4338360 4338360] [4306680 4306680] [4274280 4274280] [4241880 4241880] [4209480 4209480] [4177080 4177080] [4144680 4144680] [4112280 4112280] [4079880 4079880] [4047120 4047120] [4014360 4014360] [3981600 3981600] [3948840 3948840] [3916080 3916080] [3883320 3883320] [3850560 3850560] [3817080 3817080] [3783600 3783600] [3750120 3750120] [3716640 3717144] [3683160 3683664] [3649680 3650184] [3616200 3616704] [3582720 3583224] [3549240 3549744] [3515760 3516264] [3482280 3482784] [3448800 3449304] [3415320 3415824] [3381840 3382344] [3348360 3348864] [3314880 3315384] [3281400 3281904] [3247920 3248424] [3214440 3214944] [3180960 3181464] [3147480 3147984] [3114000 3114504] [3080520 3081024] [3047040 3047544] [3013560 3014064] [2980080 2980584] [2946600 2947104] [2913120 2913624] [2879640 2880144] [2846160 2846664] [2812680 2813184] [2779200 2779704] [2745720 2746224] [2712240 2712744] [2678760 2679264] [2645280 2645784] [2611800 2612304] [2578320 2578824] [2544840 2545344] [2511360 2511864] [2477880 2478384] [2444400 2444904] [2410920 2411424] [2377440 2377944] [2343960 2344464] [2310480 2310984] [2277000 2277504] [2243520 2244024] [2210040 2210544] [2176200 2176704] [2142360 2142864] [2108520 2109024] [2074680 2075184] [2040840 2041344] [2007000 2007504] [1973160 1973664] [1939320 1939824] [1905480 1905984] [1871640 1872144] [1837800 1838304] [1803960 1804464] [1770120 1770624] [1736280 1736784] [1702896 1703400] [1669056 1669560] [1635360 1635840] [1601808 1602288] [1568424 1568904] [1535208 1535688] [1502160 1502616] [1469256 1469712] [1436520 1436976] [1403928 1404384] [1371504 1371960] [1339248 1339704] [1307136

1307592] [1275192 1275648] [1243392 1243848] [1211736 1212192] [1180248 1180704] [1148904 1149360] [1117728 1118160] [1087272 1087128] [1056384 1056240] [1026240 1025496] [995688 994944] [965880 965136] [936240 934920] [906768 905472] [877464 876168] [10201338 10200366] [10201338 10200366] [10201338 10200366] [10201338 10200366] [10201338 10200366] [10201338 10200366] [10201338 10200366] [10201338 10200366] [10167498 10166526] [10133658 10132686] [10099818 10098846] [10065978 10065006] [10032138 10031166] [9998298 9997326] [9964458 9963486] [9930618 9929646] [9896778 9895806] [9862938 9861966] [9829098 9828126] [9795258 9794286] [9761418 9760446] [9727578 9726606] [9693738 9692766] [9659898 9658926] [9626058 9625086] [9592218 9591246] [9558378 9557406] [9524538 9523566] [9490698 9489726] [9456858 9455886] [9423018 9422046] [9389178 9388206] [9355338 9354366] [9321498 9320526] [9287658 9286686] [9253818 9252846] [9219978 9219006] [9186138 9185166] [9152298 9151326] [9118458 9117486] [9084618 9083646] [9050778 9049806] [9016938 9015966] [8983098 8982126] [8949258 8948286] [8915418 8914446] [8881578 8880606] [8847738 8846766] [8813898 8812926] [8780058 8779086] [8746218 8745246] [8712378 8711406] [8678538 8677566] [8644698 8643726] [8610858 8609886] [8577018 8576046] [8543178 8542206] [8509338 8508366] [8475498 8474526] [8441658 8440686] [8407818 8406846] [8373978 8373006] [8340138 8339166] [8306298 8305326] [8272458 8271486] [8238618 8237646] [8204778 8203806] [8170938 8169966] [8137098 8136126] [8103258 8102286] [8069418 8068446] [8035578 8034606] [8001738 8000766] [7967898 7966926] [7934058 7933086] [7900218 7899246] [7866378 7865406] [7832538 7831566] [7799058 7798086] [7765578 7764606] [7732098 7731126] [7698618 7697646] [7665138 7664166] [7631658 7630686] [7598178 7597206] [7564698 7563726] [7531218 7530246] [7497738 7496766] [7464258 7463286] [7430778 7429806] [7397298 7396326] [7363818 7362846] [7330338 7329366] [7296858 7295886] [7263378 7262406] [7229898 7228926] [7196418 7195446] [7162938 7161966] [7129458 7128486] [7095978 7095006] [7062498 7061526] [7029018 7028046] [6995538 6994566] [6962058 6961086] [6928578 6927606] [6895098 6894126] [6861978 6861006] [6828858 6827886] [6795738 6794766] [6762618 6761646] [6729498 6728526] [6696378 6695406] [6663258 6662286] [6630858 6629886] [6598458 6597486] [6566058 6565086] [6533658 6532686] [6501258 6500286] [6468858 6467886] [6436458 6435486] [6404058 6403086] [6371658 6370686] [6339258 6338286] [6306858 6305886] [6274458 6273486] [6242058 6241086] [6209658 6209790] [6177258 6177390] [6144858 6144990] [6112458 6112590] [6080058 6080190] [6047658 6047790] [6015258 6015390] [5982858 5982990] [5950098 5950230] [5917338 5917470] [5884578 5884710] [5851818 5851950] [5819058 5819190] [5786298 5786430] [5753658 5753670] [5720898 5720910] [5688138 5688150] [5655378 5655390] [5622618 5622630] [5589858 5589870] [5557098 5557110] [5524338 5524350] [5491218 5491230] [5458098 5458110] [5424978 5424990] [5391858 5391870] [5358738 5358750] [5325618 5325630] [5292498 5292510] [5259378 5259390] [5226258 5226270] [5193138 5193150] [5160018 5160030] [5126898 5126910] [5093778 5093790] [5060658 5060670] [5027538 5027550] [4994418 4994430] [4961298 4961310] [4928178 4928190] [4895058 4895070] [4861938 4861950] [4828818 4828830] [4796058 4796070] [4763298 4763310] [4730538 4730550] [4697778 4697790] [4665018 4665030] [4632258 4632270] [4599498 4599510] [4566738 4566750] [4533978 4533990] [4501218 4501230] [4468458 4468470] [4435698

4435710] [4402938 4402950] [4370178 4370190] [4337418 4337430] [4304658 4304670] [4271898 4271910] [4239138 4239150] [4206378 4206390] [4173618 4173630] [4140858 4140870] [4108098 4108110] [4075338 4075350] [4042578 4042590] [4009818 4009830] [3977058 3977070] [3944298 3944310] [3911538 3911550] [3879138 3879150] [3846738 3846750] [3814338 3814350] [3781938 3781950] [3749538 3749550] [3717138 3717150] [3684738 3684750] [3652338 3652350] [3619938 3619950] [3587538 3587550] [3555138 3555150] [3522738 3522750] [3490338 3490350] [3457938 3457950] [3425538 3425550] [3393138 3393150] [3360738 3360750] [3328338 3328350] [3295938 3295950] [3263538 3263550] [3231138 3231150] [3198378 3198390] [3165618 3165630] [3132858 3132870] [3100098 3100110] [3067338 3067350] [3034578 3034590] [3001818 3001830] [2968698 2968710] [2935578 2935590] [2902458 2902470] [2869338 2869350] [2836218 2836230] [2803098 2803110] [2769978 2769990] [2736858 2736870] [2703738 2703750] [2670618 2670630] [2637498 2637510] [2604378 2604390] [2571258 2571270] [2538138 2538150] [2505378 2505390] [2472618 2472630] [2439858 2439870] [2407098 2407110] [2374338 2374350] [2341578 2341590] [2308818 2308830] [2276418 2276430] [2244018 2244030] [2211618 2211630] [2179218 2179230] [2146818 2146830] [2114418 2114430] [2082018 2082030] [2049618 2049630] [2017218 2017230] [1984818 1984830] [1952418 1952430] [1920018 1920030] [1887618 1887630] [1855218 1855230] [1822458 1822470] [1789698 1789710] [1756938 1756950] [1724538 1724550] [1692282 1692294] [1660170 1659558] [10788368 10787909] [10788368 10787909] [10788368 10787909] [10788368 10787909] [10788368 10787909] [10788368 10787909] [10788368 10787909] [10754888 10754429] [10721408 10720949] [10687928 10687469] [10654448 10653989] [10620968 10620509] [10587488 10587029] [10554008 10553549] [10520528 10520069] [10487048 10486589] [10453568 10453109] [10420088 10419629] [10386608 10386149] [10353128 10352669] [10319648 10319189] [10285808 10285349] [10251968 10251509] [10218128 10217669] [10184288 10183829] [10150448 10149989] [10116608 10116149] [10082768 10082309] [10048928 10048469] [10015088 10014629] [9981248 9980789] [9947408 9946949] [9913568 9913109] [9879728 9879269] [9845888 9845429] [9812048 9811589] [9778208 9777749] [9744368 9743909] [9710528 9710069] [9676688 9676229] [9642848 9642389] [9609008 9608549] [9575528 9575069] [9542048 9541589] [9508568 9508109] [9475088 9474629] [9441608 9441149] [9408128 9407669] [9374648 9374189] [9341168 9340709] [9307688 9307229] [9274208 9273749] [9240728 9240269] [9207248 9206789] [9173768 9173309] [9140288 9139829] [9106448 9105989] [9072608 9072149] [9038768 9038309] [9004928 9004469] [8971088 8970629] [8937248 8936789] [8903408 8902949] [8882048 8869109] [8848208 8835269] [8814368 8801429] [8780528 8767589] [8746688 8733749] [8712848 8699909] [8679008 8666069] [8645168 8632229] [8611328 8598389] [8577488 8564549] [8543648 8530709] [8509808 8496869] [8475968 8463029] [8442128 8429189] [8408288 8395349] [8374448 8361509] [8340608 8327669] [8306768 8293829] [8272928 8259989] [8239088 8226149] [8205248 8192309] [8171408 8158469] [8137568 8124629] [8103728 8090789] [8069888 8056949] [8036048 8023109] [8002208 7989269] [7968368 7955429] [7934528 7921589] [7900688 7887749] [7866848 7853909] [7833008 7820069] [7799168 7786229] [7765328 7752389] [7731488 7718549] [7697648 7684709] [7663808 7650869] [7629968 7617029] [7596128 7583189] [7562288 7549349] [7528448 7515509] [7494608 7481669]

[7460768 7447829] [7426928 7424981] [7393088 7391141] [7359248 7357301]
[7325408 7323461] [7291568 7289621] [7257728 7255781] [7223888 7221941]
[7190048 7188101] [7156208 7154261] [7122368 7120421] [7088528 7086581]
[7054688 7052741] [7020848 7018901] [6987008 6985061] [6953168 6951221]
[6919328 6917381] [6885488 6883541] [6851648 6849701] [6817808 6815861]
[6783968 6782021] [6750128 6748181] [6716288 6714341] [6682448 6680501]
[6648608 6646661] [6614768 6612821] [6580928 6578981] [6547088 6545141]
[6513248 6511301] [6479408 6477461] [6445568 6443621] [6411728 6409781]
[6377888 6375941] [6344048 6342101] [6310208 6308261] [6276368 6274421]
[6242528 6240581] [6208688 6206741] [6174848 6172901] [6141008 6139061]
[6107168 6105221] [6073328 6071381] [6039488 6037541] [6005648 6003701]
[5971808 5969861] [5937968 5936021] [5904128 5902181] [5870288 5868341]
[5836448 5834501] [5802968 5801021] [5769488 5767541] [5736008 5734061]
[5702528 5700581] [5669048 5667101] [5635568 5633621] [5602088 5600141]
[5568608 5566661] [5535128 5533181] [5501648 5499701] [5468168 5466221]
[5434688 5432741] [5401208 5399261] [5367728 5365781] [5334608 5332661]
[5301488 5299541] [5268368 5266421] [5235248 5233301] [5202128 5200181]
[5169008 5167061] [5135888 5133941] [5135888 5133941] [5103128 5101181]
[5070368 5101181] [5037608 5068421] [5004848 5035661] [4972088 5002901]
[4939328 4970141] [4939328 4937381] [4906568 4904621] [4906568 4871861]
[4873808 4839101] [4841048 4839101] [4808288 4806341] [4775528 4806341]
[4742768 4806341] [4710008 4773581] [4677248 4740821] [4677248 4708061]
[4644488 4675301] [4611728 4642541] [4578968 4609781] [4546208 4577021]
[4513448 4544261] [4480688 4511501] [4447928 4478741] [4415168 4445981]
[4382408 4413221] [4382408 4380461] [4349288 4347341] [4316168 4314221]
[4283048 4281101] [4249928 4247981] [4216808 4214861] [4183688 4181741]
[4150568 4148621] [4117448 4115501] [4084328 4082381] [4051208 4049261]
[4018088 4016141] [3984968 3983021] [3951848 3949901] [3918728 3916781]
[3885248 3883301] [3851768 3849821] [3818288 3816341] [3784808 3782861]
[3751328 3749381] [3717848 3715901] [3684368 3682421] [3651248 3649301]
[3618128 3616181] [3585008 3583061] [3551888 3549941] [3518768 3516821]
[3485648 3483701] [3452528 3450581]] [[6866640 6866640] [6834600 6834600]
[6802560 6802560] [6770520 6770520] [6738480 6738480] [6706440 6706488]
[6674400 6674448] [6642360 6642408] [6610680 6610728] [6579000 6579048]
[6547320 6547368] [6515640 6515688] [6483960 6484008] [6452280 6452328]
[6420600 6420648] [6388920 6388968] [6357240 6357288] [6325560 6325608]
[6293880 6293928] [6262200 6262248] [6230520 6230568] [6198840 6198888]
[6166440 6166488] [6134040 6134088] [6101640 6101688] [6069240 6069288]
[6036840 6036888] [6004440 6004488] [5972040 5972088] [5939280 5939328]
[5906520 5906568] [5873760 5873808] [5841000 5841048] [5808240 5808288]
[5775480 5775528] [5742720 5742768] [5709240 5709288] [5675760 5675808]
[5642280 5642328] [5608800 5608848] [5575320 5575368] [5541840 5541888]
[5508360 5508408] [5474880 5474928] [5441400 5441448] [5407920 5407968]
[5374440 5374488] [5340960 5341008] [5307480 5307528] [5274000 5274048]
[5240520 5240568] [5207040 5207088] [5173560 5173608] [5140080 5140128]
[5106600 5106648] [5073120 5073168] [5039640 5039688] [5006160 5006208]
[4972680 4972728] [4939200 4939248] [4905720 4905768] [4872240 4872288]
[4838760 4838808] [4805280 4805328] [4771800 4771848] [4738320 4738368]
[4704840 4704888] [4671360 4671408] [4637880 4637928] [4604400 4604448]

[4570920 4570968] [4537440 4537488] [4503960 4504008] [4470480 4470528]
[4437000 4437048] [4403520 4403568] [4370040 4370088] [4336560 4336608]
[4303080 4303128] [4269600 4269648] [4236120 4236168] [4202640 4202688]
[4169160 4169208] [4135680 4135728] [4102200 4102248] [4068360 4068408]
[4034520 4034568] [4000680 4000728] [3966840 3966888] [3933000 3933048]
[3899160 3899208] [3865320 3865368] [3831528 3831528] [3797688 3797688]
[3763848 3763848] [3730008 3730008] [3696168 3696168] [3662328 3662328]
[3628488 3628488] [3594648 3594648] [3560808 3560808] [3526968 3526968]
[3493128 3493128] [3459288 3459288] [3425448 3425448] [3391608 3391608]
[3357768 3357768] [3323928 3323928] [3290088 3290088] [3256248 3256248]
[3222408 3222408] [3188568 3188568] [3154728 3154728] [3120888 3120888]
[3087048 3087048] [3053208 3053208] [3019368 3019368] [2985528 2985528]
[2951688 2951688] [2917848 2917848] [2884008 2884008] [2850168 2850168]
[2816328 2816328] [2782488 2782488] [2748648 2748648] [2714808 2714808]
[2680968 2680968] [2647128 2647128] [2613288 2613288] [2579448 2579448]
[2545608 2545608] [2511768 2511768] [2477928 2477928] [2444088 2444088]
[2410248 2410248] [2376408 2376408] [2342568 2342568] [2308728 2308728]
[2274888 2274888] [2241048 2241048] [2207208 2207208] [2173368 2173368]
[2139528 2139528] [2105688 2105688] [2071848 2071848] [2038008 2038008]
[2004168 2004168] [1970328 1970328] [1936488 1936488] [1902648 1902648]
[1868808 1868808] [1834968 1834968] [1801464 1801464] [1768104 1768104]
[1734264 1734264] [1700568 1700568] [1667040 1667040] [1633656 1633656]
[1600440 1600440] [1567368 1567368] [1534440 1534440] [1501680 1501680]
[1469064 1469064] [1436616 1436616] [1404312 1404312] [1372152 1372152]
[1340136 1340136] [1308288 1308288] [7902936 7902936] [7902936 7902936]
[7902936 7902936] [7902936 7902936] [7902936 7902936] [7902936 7902936]
[7902936 7902936] [7902936 7902936] [7869096 7869096] [7835256 7835256]
[7801416 7801416] [7767576 7767576] [7733736 7733736] [7699896 7699896]
[7666056 7666056] [7632216 7632216] [7598376 7598376] [7564536 7564536]
[7530696 7530696] [7496856 7496856] [7463016 7463016] [7429176 7429176]
[7395336 7395336] [7361496 7361496] [7327656 7327656] [7293816 7293816]
[7259976 7259976] [7226136 7226136] [7192296 7192296] [7158456 7158456]
[7124616 7124616] [7090776 7090776] [7056936 7056936] [7023096 7023096]
[6989256 6989256] [6955416 6955416] [6921936 6921936] [6888456 6888456]
[6854976 6854976] [6821496 6821496] [6788016 6788016] [6754536 6754536]
[6721056 6721056] [6687576 6687576] [6654096 6654096] [6620616 6620616]
[6587136 6587136] [6553656 6553656] [6520176 6520176] [6486696 6486696]
[6453216 6453216] [6419736 6419736] [6386256 6386256] [6352776 6352776]
[6319296 6319296] [6285816 6285816] [6252336 6252336] [6218856 6218856]
[6185376 6185376] [6151896 6151896] [6118416 6118416] [6084936 6084936]
[6051456 6051456] [6017976 6017976] [5984856 5984856] [5951736 5951736]
[5918616 5918616] [5885496 5885496] [5852376 5852376] [5819256 5819256]
[5786136 5786136] [5753736 5753736] [5721336 5721336] [5688936 5688936]
[5656536 5656536] [5624136 5624136] [5591736 5591736] [5559336 5559336]
[5526936 5526936] [5494536 5494536] [5462136 5462136] [5429736 5429736]
[5397336 5397336] [5364936 5364936] [5332536 5332536] [5300136 5300136]
[5267736 5267736] [5235336 5235336] [5202936 5202936] [5170536 5170536]
[5138136 5138136] [5105736 5105736] [5072976 5072976] [5040216 5040216]
[5007456 5007456] [4974696 4974696] [4941936 4941936] [4909176 4909176]

[4876416 4876416] [4843656 4843656] [4810896 4810896] [4778136 4778136] [4745376 4745376] [4712616 4712616] [4679856 4679856] [4647096 4647096] [4613976 4613976] [4580856 4580856] [4547736 4547736] [4514616 4514616] [4481496 4481496] [4448376 4448376] [4415256 4415256] [4382136 4382136] [4349016 4349016] [4315896 4315896] [4282776 4282776] [4249656 4249656] [4216536 4216536] [4183416 4183416] [4150296 4150296] [4117176 4117176] [4084056 4084056] [4050936 4050936] [4017816 4017816] [3985128 3984696] [3952008 3951576] [3919248 3918816] [3886488 3886056] [3853728 3853296] [3820968 3820536] [3788208 3787776] [3755448 3755016] [3722688 3722256] [3689928 3689496] [3657168 3656736] [3624408 3623976] [3591648 3591216] [3558888 3558456] [3526128 3525696] [3493368 3492936] [3460608 3460176] [3427848 3427416] [3395088 3394656] [3362328 3361896] [3329568 3329136] [3296808 3296376] [3264048 3263616] [3231288 3230856] [3198528 3198096] [3165768 3165336] [3133008 3132576] [3100248 3099816] [3067488 3067056] [3034728 3034296] [3002328 3001896] [2969928 2969496] [2937528 2937096] [2905128 2904696] [2872728 2872296] [2840328 2839896] [2807928 2807496] [2775528 2775096] [2743128 2742696] [2710728 2710296] [2678328 2677896] [2645928 2645568] [2613528 2613168] [2581128 2580768] [2548728 2548368] [2516328 2515968] [2483928 2483568] [2451528 2451168] [2419128 2418768] [2386728 2386368] [2354328 2353968] [2321568 2321208] [2288808 2288448] [2256048 2255688] [2223288 2222928] [2190528 2190168] [2157768 2157408] [2125008 2124648] [2091888 2091528] [2058768 2058408] [2025648 2025288] [1992528 1992168] [1959408 1959048] [1926288 1925928] [1893168 1892808] [1860048 1859688] [1826928 1826568] [1794168 1793808] [1761552 1761192] [1728456 1728096] [1695504 1695144] [1662696 1662336] [1630392 1630032] [1598232 1597896] [1566216 1565904] [1534344 1534056] [1502616 1502352] [1471032 1470768] [1439592 1439328] [1408656 1408392] [1377840 1377576] [1347744 1346904] [1317792 1316376] [1287384 1285968] [1257120 1255704] [1226976 1225584] [1196976 1195584] [1167672 1165728] [1138512 1135992] [1108896 1106952] [1079424 1078056] [1050096 1048752] [1020936 1020144] [991608 991368] [962760 962760] [933744 934296] [905424 905424] [877272 877272] [849264 849264] [821400 821400] [793368 793368] [765504 765504] [737256 737256] [709152 709152] [681168 681168] [653328 653328] [625608 626136] [597792 598248] [570552 571008] [543432 543360] [515928 516168] [488544 488784] [461280 461520] [434136 434376] [407640 407376] [380760 380496] [354504 353736] [328368 327096] [301848 300576] [275952 274176] [249672 247896] [223224 221472] [196896 195168] [170688 168984] [145104 143400] [119640 117936] [93792 92592] [68568 67368] [43464 42264] [18456 17280]

[18456 17280] [6935562 6934680] [6935562 6934680] [6935562 6934680] [6935562 6934680] [6935562 6934680] [6935562 6934680] [6935562 6934680] [6935562 6934680] [6902442 6901560] [6869322 6868440] [6836202 6835320] [6803082 6802200] [6769962 6769080] [6736842 6735960] [6723042 6702840] [6690282 6670080] [6690282 6637320] [6657522 6604560] [6624762 6571800] [6624762 6539040] [6592002 6506280] [6559242 6473520] [6526482 6440760] [6526482 6408000] [6493722 6408000] [6460962 6375240] [6428202 6375240] [6395442 6342480] [6362682 6342480] [6329922 6342480] [6297162 6309720] [6264402 6309720] [6231642 6276960] [6198882 6244200] [6166122 6211440] [6133362 6178680] [6100602 6145920] [6100602 6113160] [6067842 6080400] [6035082 6047640] [6002322 6014880] [5969562 5982120] [5969562 5949360] [5936442 5916240] [5903322 5883120] [5870202 5850000] [5837082 5816880] [5803962 5783760] [5770842 5750640] [5737722 5737200] [5704602 5704080] [5671482 5670960] [5638362 5637840] [5605242 5604720] [5572122 5571600] [5539002 5538480] [5505882 5505360] [5472402 5471880] [5438922 5438400] [5405442 5404920] [5371962 5371440] [5338482 5337960] [5305002 5304480] [5271522 5271000] [5238402 5237880] [5205282 5204760] [5172162 5171640] [5139042 5138520] [5105922 5105400] [5072802 5072280] [5039682 5039160]]];

ea = [[<1 0 0 0> <1 126 133 7> <1 392 399 7> <1 624 624 0>] [<1 0 0 0> <1 168 175 7> <1 553 560 7> <1 624 624 0>]];

p2 = [[[1335 1335] [1335 1335] [1335 1335] [1335 1335] [1335 1335] [1335 1335] [1335 1335] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1395 1395] [1395 1395] [1395 1395] [1395 1374] [1395 1395] [1395

1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1391 1391] [1410 1410] [1404 1405] [1398 1398] [1391 1391] [1384
1384] [1377 1378] [1371 1371] [1364 1364] [1358 1358] [1351 1351] [1344 1344]
[1338 1338] [1331 1331] [1325 1325] [1319 1319] [1312 1312] [1306 1306] [1299
1300] [1269 1293] [1287 1287] [1256 1281] [1273 1273] [1242 1242] [1235 1259]
[1228 1227] [1221 1221] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1380
1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380]
[1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350
1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350]
[1350 1304] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350
1350] [1350 1350] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1360 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365
1365] [1365 1365] [1365 1365] [1380 1380] [1380 1380] [1380 1380] [1380 1380]
[1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380
1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380]
[1380 1380] [1380 1380] [1380 1380] [1380 1380] [1365 1365] [1365 1365] [1365
1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365
1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1350
1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350]
[1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350
1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350]
[1350 1350] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365
1365] [1365 1365] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380]
[1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380
1380] [1380 1380] [1380 1380] [1365 1365] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1365 1365] [1365 1365] [1350 1350] [1350 1350] [1350 1350] [1350
1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350]
[1350 1350] [1350 1350] [1350 1350] [1350 1350] [1365 1365] [1365 1365] [1365
1365] [1350 1350] [1344 1344] [1338 1364] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [0
0] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]

[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [890 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 952] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1380 1380]
[1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [0 0]
[1365 1365] [1365 0] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [0 1365]
[1365 1365] [0 1365] [1365 1365] [1365 0] [1365 1365] [1365 0] [1365 0] [1365
1365] [1365 1365] [0 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [0 1365] [1380
1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380]
[1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380
1380] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380
1380] [1380 1380]] [[1335 1335] [1335 1335] [1335 1335] [1335 1335] [1335 1333]
[1335 1335] [1335 1335] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320
1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320] [1320 1320]
[1320 1320] [1320 1320] [1320 1320] [1350 1350] [1350 1350] [1350 1350] [1350
1350] [1350 1350] [1350 1350] [1350 1350] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1365 1365] [1365 1365] [1365 1365] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1408 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410

1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1396 1396] [1390 1390] [1410 1410] [1404
1404] [1397 1397] [1391 1391] [1384 1384] [1378 1378] [1372 1372] [1365 1365]
[1359 1359] [1352 1352] [1346 1346] [1340 1340] [1334 1334] [1327 1327] [0 0] [0
0] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410
1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410]
[1410 1410] [1410 1410] [1410 1410] [1410 1410] [1410 1410] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395]
[1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395
1395] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380]
[1380 1380] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350
1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350]
[1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350
1350] [1350 1350] [1350 1350] [1365 1365] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365
1365] [1365 1365] [1365 1365] [1365 1365] [1380 1380] [1380 1380] [1380 1380]
[1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380
1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380]
[1380 1380] [1380 1380] [1380 1380] [1362 1380] [1380 1380] [1365 1365] [1365
1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365
1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365]
[1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365
1365] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350]
[1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1347] [1350
1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350] [1350 1350]
[1350 1350] [1350 1350] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365
1365] [1365 1365] [1365 1365] [1380 1380] [1380 1380] [1380 1380] [1380 1380]
[1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1365 1365] [1359
1359] [1379 1379] [1373 1373] [1367 1367] [1346 1346] [1340 1339] [1334 1333]
[1328 1327] [1322 1321] [1316 1316] [1310 1310] [1289 1289] [1284 1284] [1254
1278] [1248 1272] [1267 1267] [1261 1261] [1256 1255] [1250 1250] [1221 1244]
[1215 1239] [1234 1210] [1228 1204] [1222 1221] [1215 1192] [1222 1199] [1202
1192] [1209 1186] [1180 1203] [1173 1173] [1167 1167] [1161 1161] [1168 1168]
[1161 1161] [1177 1177] [1171 1171] [1166 1166] [1160 1160] [1155 1133] [1159
1162] [1135 1135] [1130 1152] [1146 1133] [1141 1141] [1136 1136] [1131 1131]
[1104 1125] [1120 1120] [1094 1115] [1089 1110] [1105 1105] [1079 1100] [1095
1095] [1102 1101] [1097 1096] [1092 1091] [1066 1066] [1061 1061] [1077 1056]
[1051 1051] [1046 1046] [1042 1041] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [0 0] [0

0] [0 0] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [575 1380] [1365 1365] [0 1365] [1365 1365] [1365 1365] [0 1365] [1365 1365] [1365 1365] [1365 1365] [0 1365] [1365 0] [1365 1365] [1365 0] [1365 1365] [1365 0] [1365 0] [1365 1365] [1365 0] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [0 1365] [1365 1365] [1365 1365] [1365 1365] [1365 1365] [0 1365] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 560] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1395 1395] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380] [1380 1380]]];

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

Kısıt programlama ile çizelgeleme problemlerinin nasıl çözüldüğünün anlatılması için kısıt programlamadan, uygulama alanlarından ve diğer algoritmalar ile kıyaslamalarından bahsedilmiş ve çizelgele hakkında bilgiler verilmiştir.

Üçüncü bölümde IBM ILOG CP eniyileme programından bahsedilmiş ve kullanımı konusunda bilgiler verilmiştir. Tüm bu anlatılanlar uygulama problemi çözülerek pekiştirilmiştir.

Uygulama probleminde yüksek ölçekli bir çizelgeleme problemindeki tesislerin bakım zamanları ve her bakımda yüklenmesi gereken güç miktarları minumum maliyetleri ile bulunmuş ve tüm bunlar yapılırken müşteri talebinin karşılanması sağlanmıştır. Birinci çözüm yolu olarak senaryolar bütün olarak düşünülmüş ve çözülmüştür. Bu durumda problemin çözümü yavaşlamıştır. İkinci bir çözüm olarak ise senaryolar ayrı olarak çözülmüş ve en sonda birleştirilmiştir. Bu durumda çözüm ortalama üçte bir oranında daha hızlı elde edilmiştir.

Sonuç olarak kısıt programlama ile çizelgeleme problemlerinin nasıl çözüldüğü ve örnek bir uygulama detayları ile anlatılmıştır.

KAYNAKLAR

[1] IBM, Detailed Scheduling in IBM ILOG OPL with IBM ILOG CP Optimizer, Tutorial, 2009. p. 43.

[2] FROMHERZ, M.P.J., Constraint –based Scheduling, American Control Conference (ACC'01), Arlington, VA, June 2001. p. 14.

[3] LUSTIG I.J, VE PUGET J.F., Program != Program: Constraint Programming and its Relationship to Mathematical Programming. Ilog, 1999.

[4] BAPTISTE P., LE PAPE C., VE NUIJTEN W., "Incorporating Efficient Operations Research Algorithms in Constraint-based Scheduling." In: First Int. Joint Workshop on Artificial Intelligence and Operations Research, Timberline Lodge, Oregon, 1995.

[5] BAPTISTE P., LABORIE P., LE PAPE C., NUIJTEN W., Constraint –based Scheduling and Planning, Handbook of Constraint Programming, 2006, pp: 761 – 799.

[6] STEELE G.R., JR. The Definition and Implementation of a Computer Programming Language Based on Constraints. PhD thesis, Massachusetts Institute of Technology, 1980.

[7] BOCKMAYR A., HOOKER J.N., Constraint Programming, K. Aardal et al., Eds., Handbooks in OR & MS, Vol. 12, 2005. pp: 559 – 600.

[8] ALTHAUS, E., BOCKMAYR A., ELF M., KASPER T., €UNGER M.J., MEHLHORN K., SCIL-Symbolic constraints in integer linear programming. 10th European Symposium on Algorithms, ESA' 02, Springer, Rome, LNCS 2461, 2002. pp. 75–87.

[9] HOOKER, J. N, A framework for integrating solution methods, in: H. K. Bhargava, Mong Ye (eds.), Computational Modeling and Problem Solving in the Networked World (Proceedings of ICS 2003), Kluwer, 2003, pp. 3–30.

[10] OTTOSSON, G., E. THORSTEINSSON, Linear relaxations and reduced-cost based propagation of continuous variable subscripts. Second International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, CPAIOR2000, University of Paderborn.

[11] CARLSSON M., OTTOSSON G., VE CARLSON B., "An Open-Ended Finite Domain Constraint Solver." In: Int. Symp. on Programming Languages: Implementations, Logics, and Programming (PLILP'97), Springer-Verlag, LNCS 1292, Sept. 1997.

[12] MACKWORTH A. K., "Consistency in Networks of Relations." In:AIJournal,vol.8,no.1,1977,pp.99-118.

[13] TSANG E., Foundations of Constraint Satisfaction. Academic Press, San Diego, California, 1993.

[14] BACCHUS F. VE VAN RUN P., "Dynamic Variable Ordering in CSPs." In: Principles and Practice of Constraint Programming (CP'95), Springer-Verlag, LNCS976,Sept.1995,pp.258-275.

[15] GETOOR L., OTTOSSON G., FROMHERZ M. P. J., VE CARLSON B., "Effective Redundant Constraints for Online Scheduling." In: AAAI'97, Providence, RI, July 1997, pp. 302-307.

[16] HOGG T., HUBERMAN B., VE WİLLİAMS C., "Phase Transition and the Search Problem." In: AI Journal, vol. 81, 1996, pp. 1-15.

[17] NADEL B., "Tree Search and Arc Consistency in Constraint Satisfaction." In: Search in Artificial Intelligence, Springer-Verlag, 1988, pp. 287-342.

[18] KUMAR V., "Algorithms for Constraint Satisfaction Problems: A Survey." In: AI Magazine, vol. 13, no. 1, 1992, pp. 32-44.

[19] HARVEY W. D. VE GINSBERG M. L., "Limited Discrepancy Search." In: IJCAI-95, Montreal, Quebec, 1995, pp. 607-613.

[20] WALSH T., "The Constrainedness Knife-Edge." In: AAAI'98, 1998, pp. 406-411.

[21] BAPTISTE P., LE PAPE C., VE NUIJTEN W., "Constraint-based Optimization and Approximation for Job-shop Scheduling." In: AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems, IJCAI'95, Montreal, Canada, 1995.

[22] GRAHAM R. L., LAWLER E. L., LENSTRA J. K., VE RİNNYOO KAN A. H. G., "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey." In: Discrete Optimization 1977, Vancouver, BC, 1977.

[23] GAREY M. R. VE JOHNSON D. S., Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., New York, 1979.

[24] BŁAZEWİCZ J., ECKER K., SCHMİDT G., VE WEGLARZ J., Scheduling in Computer and Manufacturing Systems. Springer-Verlag, 1993.

[25] ZWEBEN M. VE FOX M., Intelligent Scheduling. Morgan Kaufman, 1994.

[26] SIMONIS H., "Scheduling and Planning with Constraint Logic Programming." In: Practical Applications of Constraint Technology, Tutorial, Paris, France, 1995.

[27] FROMHERZ M. P. J. VE CONLEY J. H., "Issues in Reactive Constraint Solving." In: Workshop on Concurrent Constraint Programming for Time Critical Applications — COTIC'97, CP'97, Linz, Austria, Oct. 1997.

[28] 142.ibm.com

EKLER

EK A

begin main

timesteps 623

weeks 89

campaigns 2

scenario 2

epsilon 0.01

powerplant1 1

powerplant2 2

constraint13 4

constraint14 1

constraint15 0

constraint16 0

constraint17 0

constraint18 0

constraint19 0

constraint20 0

constraint21 0

durations 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24

24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24

demand 51318.22 49846.98 51838.91 51954.74 51179.61 50547.53 51042.99 51271.81 49524.09 54698.31 54744.59 55471.89 54682.56 55446.56 52487.78 50128.80 53525.32 55031.02 55385.47 55227.54 55521.91 55325.86 53172.85 56015.75 56879.57 59438.39 58819.93 58532.62 55484.21 53291.11 58617.69 58388.82 60185.93 58212.23 59034.29 56331.26 53832.69 59471.54 58397.32 59757.90 61104.82 62377.78 60361.66 62875.97 61598.06 62856.89 62627.67 63725.55 62756.30 61584.93 59219.11 62701.04 62165.10 65653.74 66046.71 68562.75 63035.71 59196.14 65293.80 64642.06 63605.13 63094.14 56830.21 61267.11 59103.30 64456.40 65815.87 64827.48 64213.95 63627.87 60596.89 59565.71 67726.86 68411.49 71135.58 69738.95 71949.96 68919.68 66882.10 72812.30 73088.01 72989.55 72559.48 75673.61 68766.00 67011.61 74393.41 75998.29 76563.03 73832.33 74450.09 70083.55 66858.46 72274.05 76364.39 75657.34 74114.63 76709.51 69843.00 65853.91 69789.38 66676.48 69345.31 74101.79 74451.42 69795.29 67900.98 72194.49 72349.19 71119.26 71693.24 69638.52 64309.63 60128.31 64481.77 70243.56 71726.93 69604.43 68904.97 63741.24 63193.69 72898.40 74093.36 75118.25 73998.44 75951.37 75541.05 73859.82 81225.73 81814.54 80937.61 79177.34 78882.25 72323.51 69058.78 73035.56 71991.77 72592.88 70811.70 69994.58 64141.31 61705.20 70024.38 69049.91 67522.53 65964.26 66091.49 60126.61 54993.01 61054.85 62523.40 64226.30 64039.36 64452.19 58672.46 56861.45 63262.33 61216.23 61001.43 60999.20 60999.20 58215.19 56327.49 58856.91 59165.74 59271.29 59453.19 59252.22 57905.75 55976.03 58544.26 58645.01 58683.83 57995.22 58356.71 55891.49 56118.30 59167.15 57960.60 57335.80 57722.10 58400.01 57492.36

56043.59 59932.88 57082.11 58982.68 57335.46 57700.45 53227.03 53433.49

58191.59 57156.67 60304.33 57631.20 56828.22 57007.24 53817.78 60048.22

59425.85 59119.17 59737.17 60474.64 53403.50 53000.75 55211.06 54166.24

54715.33 56698.07 56244.30 53594.05 51774.31 56316.92 56926.66 56951.69

56738.01 56465.44 50155.96 48568.16 48970.77 55113.84 57202.37 58946.37

55839.11 54749.83 51959.75 56790.68 57760.97 58107.35 56585.37 58368.55

51202.36 45406.66 45885.51 53296.88 56322.47 56369.38 55977.74 51844.74

49102.77 50658.82 53912.48 55143.87 51855.84 52876.93 50765.16 48363.20

52984.40 54589.52 54232.58 55860.26 54089.67 50301.78 47609.78 52676.30

52349.56 50218.67 44695.11 47283.37 46755.82 46451.17 53556.54 54128.76

53140.71 50930.56 48971.96 45378.97 42293.18 48982.63 49282.46 51217.80

51656.39 51192.77 45024.53 43990.15 49630.71 47353.19 48366.30 49619.29

49535.74 49141.96 46887.29 53397.24 53036.12 53658.69 50699.45 53739.83

48076.14 45932.99 50837.32 50751.18 51750.81 50883.69 51478.86 51290.98

47970.74 53221.52 52433.53 52733.13 53188.24 54473.40 47929.26 48126.57

50134.61 51712.42 51980.11 51248.11 49013.07 51410.81 49088.52 52326.84

53607.78 54040.95 53322.86 52781.10 52079.17 49937.01 55051.07 56166.81

56656.10 56625.15 56133.56 51932.37 48400.69 52753.16 51925.73 53487.04

52559.14 52755.21 49563.64 46652.23 50651.64 51396.00 51207.56 51378.64

52015.48 49643.78 46577.47 51089.33 48842.97 53026.47 53426.44 53780.66

50305.04 47613.19 52167.70 54435.10 54857.06 55096.00 56057.37 52681.37

49342.95 54456.70 54622.48 56698.99 55654.10 57919.52 55618.01 52796.41

55795.94 57145.03 57373.31 57566.94 55372.54 52298.68 50662.80 57032.05

57340.48 53319.64 53223.53 55343.96 51341.20 48822.98 53092.61 51887.60

53081.49 53599.51 54355.48 51106.22 49589.56 52862.14 54921.41 56120.71

54649.87 55270.29 53829.99 52085.67 57458.97 58691.37 56173.98 56215.88

56586.39 52378.88 50081.14 53951.78 54458.32 57292.46 58525.54 58667.79

54065.86 51611.01 56861.24 59331.01 60715.44 57988.30 58926.65 56603.43

56343.07 61204.36 61373.93 58167.09 61141.46 60340.68 56615.08 48732.42

56168.47 58411.57 60343.91 62177.15 63067.33 61624.65 58519.66 62210.68

62436.43 63449.47 61875.16 61921.16 58121.30 58794.82 63933.14 61652.99

60944.18 61098.07 60192.79 59231.10 58401.59 62845.74 61628.90 65113.64

62315.93 64088.49 69239.25 66131.06 68613.51 72146.10 70437.07 69858.41

70460.22 65825.42 63528.13 71071.35 73874.40 74764.75 77684.96 76299.63

71518.78 68977.81 75778.98 74283.02 73795.15 73576.55 74417.09 73656.18

68151.41 74666.53 72950.90 72486.06 71775.44 74461.47 72274.88 70418.32

71550.07 77176.81 79493.48 79763.63 76810.46 71458.30 68119.32 65443.26

70324.16 74535.09 75545.39 72679.71 68037.18 66856.05 74809.65 76959.45

77478.50 77764.44 78434.15 79044.34 78440.39 81769.61 79700.48 79519.15

79261.00 77738.48 73609.76 71914.38 75791.29 74288.32 74695.73 75635.15

74610.01 66418.56 64137.64 69631.33 71671.45 73557.16 70155.12 69775.19

62867.06 61054.36 66390.65 67348.62 66863.47 68102.76 68409.63 65326.18

63788.07 69965.49 69189.36 70835.99 68633.58 68858.60 60046.19 55195.63

63733.33 65943.89 65120.62 64282.45 66154.80 59422.07 56264.01 60737.68

59172.70 59625.73 60817.20 65720.97 64672.29 61221.71 66846.43 62922.36

62583.66 64705.32 64174.31 62522.08 62133.70 61883.71 62611.17 63798.66

62340.08 63200.01 62317.02 62218.78 68762.82 67117.21 65899.47 65476.25

69215.43 61713.10 58776.71 60997.96 61836.44 63484.99 64563.25 62835.45

59060.36 54948.30 58697.77 54890.33 56366.77 56136.64 59279.71 53593.13

51810.08 53028.98 56842.75 57294.69 55526.59 55538.79 54794.14 52153.39

56165.39 57546.71 57809.17 58240.65 57809.91 56220.87 52167.99 59105.47
60184.98 60207.44 60402.99 60418.41 55863.53 51335.57 52938.78 46601.63
52293.29 55461.60 53860.88 51365.82 48971.33 55494.81 52006.22 57155.88
57607.17 56380.36 50369.16 44768.69 53083.14 55224.46 56217.48 46663.61
50586.99 48778.21 42959.41 51530.06 51871.66 52337.00 52591.06 51268.61

demand 51629.45 48625.52 52573.78 52329.49 53132.42 53063.58 52945.79
53158.33 49950.74 52953.03 53364.78 53408.09 53789.17 54891.17 52393.59
48629.52 53998.14 53111.94 51986.66 51844.08 52312.12 50364.20 47869.28
54863.56 55742.06 56156.07 54624.91 54545.79 51128.46 49580.27 54114.77
55184.67 55074.18 54272.41 55084.89 55287.27 52995.16 56609.70 56651.33
57977.78 58832.58 58986.82 54305.23 55311.55 60255.90 60094.32 57513.05
58295.48 59488.14 60325.13 58523.05 62014.48 62473.01 65737.27 66921.20
67447.34 63716.70 58097.51 64533.28 65219.52 66908.22 64266.43 54138.35
54447.84 54118.56 64609.84 64824.71 64563.69 62148.93 60980.40 57741.96
57051.36 63706.88 61912.84 62656.63 66941.57 65062.47 61433.00 60685.75
69440.93 70375.92 69133.97 74470.85 75317.21 73818.36 69871.73 70888.99
72910.45 70518.74 72978.19 73817.22 65065.27 61413.30 69271.81 71910.54
69988.47 68793.50 70301.77 65809.23 62567.88 67448.95 69801.59 70668.13
70645.69 68485.79 62024.60 58522.12 67006.22 69997.69 73156.15 75401.37
75161.74 70180.63 67641.70 78012.62 81613.99 83214.55 80831.71 82783.97
79373.99 76747.69 82584.12 80293.78 82342.06 80299.08 81220.63 76731.50
76472.69 78175.01 78953.67 79228.76 75095.79 72211.51 66542.47 64344.19
71748.78 73684.31 73273.83 71139.34 72077.15 67291.08 66087.58 73292.15
73602.24 71611.97 73600.62 76563.67 72303.17 71744.73 72754.65 73802.71
70767.89 71281.20 69819.53 62942.46 60885.31 67328.53 69772.15 69959.72

69294.85 67016.57 56278.48 56734.06 66592.37 68480.68 65954.04 65366.11

66137.10 64181.87 62043.12 67745.09 63796.46 63637.77 63667.07 64152.97

60139.48 57821.07 63021.26 61859.38 62389.93 62988.18 60111.41 60360.38

58101.98 62233.75 61551.48 62120.01 59551.20 64052.05 60808.65 60342.95

59880.10 59948.89 62654.55 61419.63 60683.42 61786.49 57581.93 60823.09

61039.40 60782.17 65072.87 63755.95 63471.64 62139.60 61199.19 63672.61

61518.41 62629.35 62613.27 60273.31 59211.02 60537.60 61290.76 60287.95

58463.32 58641.93 55983.00 53529.20 53420.71 56521.48 58777.10 59029.84

59244.42 56093.99 53084.17 58414.56 59416.59 58316.52 58729.36 58555.51

54712.33 50455.78 51016.08 58195.11 59767.06 57883.33 60199.98 56129.51

54597.75 55375.67 56989.14 55244.85 55152.49 57472.61 55362.34 52288.33

56611.24 55577.67 56548.00 57328.90 57007.24 54245.02 50890.01 55865.52

56521.06 55344.49 50041.29 49809.23 49850.82 48016.53 54852.45 54733.85

54608.35 54296.75 53305.01 49974.74 46047.85 52364.92 50731.85 50559.16

53116.82 53047.58 49581.11 46606.91 52577.92 52233.09 54143.41 54064.43

53951.19 51981.08 49584.80 54674.77 55082.25 55313.97 54085.24 56657.91

52794.80 50812.82 53950.92 53474.62 54856.75 53365.67 52953.20 51169.34

48665.45 53727.46 53829.05 52650.39 52715.04 53912.13 52677.53 50879.02

55025.18 53668.64 53106.85 54037.46 48500.37 51621.20 49322.27 55067.63

55621.80 56538.88 57113.66 56744.08 53171.05 50247.64 56388.79 57917.45

58272.93 57928.92 57101.84 53124.44 48932.29 52628.84 52313.58 51424.95

50863.12 50492.07 47606.33 44107.90 47362.87 45937.96 47747.50 46110.07

49078.77 47942.39 44573.79 47538.70 46099.63 49708.19 50306.48 51190.57

48786.85 45370.11 51965.77 52009.85 52098.38 52720.10 52840.93 50224.68

47228.78 54005.08 55412.58 54875.59 53379.08 55089.28 51943.47 50363.61

55354.66 55125.67 55373.06 56780.38 55283.97 51934.83 49710.07 54779.70

55477.42 55380.02 57574.15 56360.64 45315.49 42973.35 50856.75 52721.65

54452.76 53908.28 54012.36 49200.08 47869.16 53675.60 54383.30 54725.67

54398.18 52109.85 50616.50 53058.44 58835.17 56396.46 58805.97 57050.67

59452.74 55854.10 53725.24 58703.66 59615.78 59918.75 60441.69 61306.98

59079.29 54288.07 60349.06 60381.06 60320.42 60477.83 60497.79 56698.49

55247.09 62794.33 62017.19 62886.95 64725.52 64550.20 63816.28 62632.31

64963.06 67940.80 66015.90 70334.69 70285.09 69820.55 66601.09 70379.61

69564.36 68410.21 68959.21 68953.17 57575.47 57277.29 67336.34 69148.09

70030.64 70684.63 68143.52 63277.84 61132.79 68197.27 68411.73 67906.67

67817.52 68245.86 63698.18 62282.44 68353.64 71278.21 72103.90 71574.35

71718.68 65558.87 66598.97 73272.85 72665.72 72158.22 69327.99 71084.50

63925.35 58277.21 65906.62 65470.35 66355.65 68230.19 69017.76 64823.39

62393.38 71266.91 68262.37 69971.01 70387.01 70530.75 67838.97 61540.49

60603.49 68674.92 71111.18 73921.50 76172.81 73681.91 68789.56 71042.43

79812.27 83028.93 84575.26 84580.94 80907.31 80139.64 82679.86 83754.62

83574.78 82639.65 81121.65 82344.33 79316.33 82103.64 82281.72 83836.78

85014.67 80898.91 72575.93 68236.78 74791.60 75710.28 71856.80 72689.41

71717.93 67431.91 66030.19 75387.76 75672.03 74995.06 78470.31 79833.26

76012.33 74718.43 77449.65 78706.43 75578.81 75170.68 75408.69 68104.20

64729.66 72580.34 72215.97 73462.46 74485.64 72828.78 66023.96 63274.56

67789.39 68099.19 69476.04 71151.51 73526.11 68461.61 66650.58 69311.02

66688.78 68918.18 67013.21 65804.26 58803.54 57132.14 65177.64 64162.17

61883.99 64361.08 66469.24 62341.39 58723.58 62956.27 63108.87 64102.00

63497.79 65588.37 62330.18 59633.22 63442.35 63108.17 64910.97 64664.74

```
66380.47  62317.19  59613.73  61774.73  59801.70  59887.04  60317.54  61565.60

64751.33  61418.63  64355.88  62210.72  63821.94  63500.38  63367.61  59473.57

57838.38  57825.37  60135.65  60695.55  59894.11  58157.07  55875.72  51506.44

56384.55  56200.63  57948.10  55303.16  55833.81  53279.29  50218.25  54623.68

56335.03  59482.89  59407.32  57316.77  57312.64  52422.48  52952.52  52082.25

57587.52  57537.36  57573.66  56675.01  52925.21  57861.07  56650.83  59720.17

59819.52  60511.60  55539.55  52237.72  58560.03  59197.27  58154.45  51829.84

51978.19 52400.22 48534.55 55700.03 57692.85 58048.94 58022.59 57073.17

end main

begin powerplant

name PowerPlant_1_0

type 1

index 0

scenario 2

timesteps 623

pmin 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

pmax 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00

100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

cost 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

```
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
```

```
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
```

```
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

pmin 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

pmax 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00
100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00

100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00
100000.00	100000.00	100000.00	100000.00	100000.00	100000.00	100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

100000.00 100000.00 100000.00 100000.00 100000.00 100000.00 100000.00

cost 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00

```
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
```

```
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
```

```
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00  10000.00
10000.00 10000.00 10000.00 10000.00 10000.00 10000.00 10000.00
```

end powerplant

begin powerplant

name PowerPlant_2_0

type 2

index 0

stock 4974480

campaigns 2

durations 5 8

current_campaign_max_modulus 176400

max_modulus 176400 176400

max_refuel 9102240 9102240

min_refuel 9102240 9102240

refuel_ratio 4 4

current_campaign_stock_threshold 1764000

stock_threshold 1764000 1764000 1764000

pmax 1335.00 1335.00 1335.00 1335.00 1335.00 1335.00 1335.00 1320.00 1320.00
1320.00 1320.00 1320.00 1320.00 1320.00 1320.00 1320.00 1320.00 1320.00
1320.00 1320.00 1320.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00

```
1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00
1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00
1395.00  1395.00  1395.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00  1410.00
1410.00  1410.00  1410.00  1410.00  1410.00  1395.00  1395.00  1395.00  1395.00
1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00
1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1395.00
1395.00  1395.00  1395.00  1395.00  1395.00  1395.00  1380.00  1380.00  1380.00
1380.00  1380.00  1380.00  1380.00  1350.00  1350.00  1350.00  1350.00  1350.00
1350.00  1350.00  1350.00  1350.00  1350.00  1350.00  1350.00  1350.00  1350.00
1350.00  1350.00  1350.00  1350.00  1350.00  1350.00  1350.00  1365.00  1365.00
1365.00  1365.00  1365.00  1365.00  1365.00  1365.00  1365.00  1365.00  1365.00
1365.00  1365.00  1365.00  1380.00  1380.00  1380.00  1380.00  1380.00  1380.00
1380.00  1380.00  1380.00  1380.00  1380.00  1380.00  1380.00  1380.00  1380.00
```

1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1365.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1350.00 1350.00
1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1380.00
1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00
1380.00 1380.00 1380.00 1380.00 1365.00 1365.00 1365.00 1365.00 1365.00
1365.00 1365.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1380.00 1380.00 1380.00 1380.00
1380.00 1380.00 1380.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1395.00 1395.00 1395.00

1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00

1395.00 1395.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00

1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00

1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00

1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00

1365.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00

1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1395.00 1395.00 1395.00

1395.00 1395.00 1395.00 1395.00 1380.00 1380.00 1380.00 1380.00 1380.00

1380.00 1380.00

max_stock_before_refueling 3175200 3175200

max_stock_after_refueling 14112000 14112000

refueling_cost 20.37 19.38

fuel_price 20.68

begin current_campaign_profile

profile_points 7

decrease_profile 1764000 1 1411200 0.95 1058400 0.9 705600 0.84 352800 0.79 0 0.74 0 0.74

end current_campaign_profile

begin profile

campaign_profile 0

profile_points 7

decrease_profile 1764000 1 1411200 0.95 1058400 0.9 705600 0.84 352800 0.79 0 0.74 0 0.74

end profile

begin profile

campaign_profile 1

profile_points 7

decrease_profile 1764000 1 1411200 0.95 1058400 0.9 705600 0.84 352800 0.79 0 0.74 0 0.74

end profile

end powerplant

begin powerplant

name PowerPlant_2_1

type 2

index 1

stock 6866640

campaigns 2

durations 9 6

current_campaign_max_modulus 183600

max_modulus 183600 183600

```
max_refuel 12484800 12484800

min_refuel 6462720 6462720

refuel_ratio 4 4

current_campaign_stock_threshold 1836000

stock_threshold 1836000 1836000 1836000

pmax 1335.00 1335.00 1335.00 1335.00 1335.00 1335.00 1335.00 1320.00 1320.00
1320.00 1320.00 1320.00 1320.00 1320.00 1320.00 1320.00 1320.00 1320.00
1320.00 1320.00 1320.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
```

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00
1410.00 1410.00 1410.00 1410.00 1410.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00
1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1380.00 1380.00 1380.00
1380.00 1380.00 1380.00 1380.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00
1365.00 1365.00 1365.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00
1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00
1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1365.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1350.00 1350.00
1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1380.00
1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00
1380.00 1380.00 1380.00 1380.00 1365.00 1365.00 1365.00 1365.00 1365.00
1365.00 1365.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00
1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1350.00 1365.00 1365.00
1365.00 1365.00 1365.00 1365.00 1365.00 1380.00 1380.00 1380.00 1380.00

1380.00 1380.00 1380.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00

1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00

1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1410.00

1410.00 1410.00 1410.00 1410.00 1410.00 1410.00 1395.00 1395.00 1395.00

1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00 1395.00

1395.00 1395.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00

1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00

1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00

1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00 1365.00

1365.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00

1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1395.00 1395.00 1395.00

1395.00 1395.00 1395.00 1395.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00 1380.00

max_stock_before_refueling 3304800 3304800

max_stock_after_refueling 14688000 14688000

refueling_cost 19.96 19.02

fuel_price 20.26

begin current_campaign_profile

profile_points 7

decrease_profile 1836000 1 1468800 0.95 1101600 0.9 734400 0.84 367200 0.79 0 0.74 0 0.74

end current_campaign_profile

begin profile

campaign_profile 0

profile_points 7

decrease_profile 1836000 1 1468800 0.95 1101600 0.9 734400 0.84 367200 0.79 0 0.74 0 0.74

end profile

begin profile

campaign_profile 1

profile_points 7

decrease_profile 1836000 1 1468800 0.95 1101600 0.9 734400 0.84 367200 0.79 0 0.74 0 0.74

end profile

end powerplant

begin constraint

type 13

index 0

powerplant 0

campaign 0

earliest_stop_time 18

latest_stop_time 26

end constraint

begin constraint

type 13

index 1

powerplant 0

campaign 1

earliest_stop_time 56

latest_stop_time 64

end constraint

begin constraint

type 13

index 2

powerplant 1

campaign 0

earliest_stop_time 24

latest_stop_time 32

end constraint

begin constraint

type 13

index 3

powerplant 1

campaign 1

earliest_stop_time 79

latest_stop_time 87

end constraint

begin constraint

type 14

index 0

set 0 1

spacing 6

end constraint

EK B

```
Imports System.IO.File

Public Class Form1

Dim DosyaKaydet As System.IO.StreamWriter

Dim DosyaOku As System.IO.StreamReader

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

Dim a, b, alltime, week, campaign, scenario, epsilon, type1, type2, demand, pmin,
pmin2, pmax, pmax2, c1, c12, firsta, att2, nrmal, att3 As String

Dim c, alltimes, nbcamp, nbscn, nbtype1, nbtype2, hj, hjj, denetleme, denetleme2,
ct13, ct14 As Integer

Dim s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12, s13, s14, s15, s16, s17, s18, points
As Integer

Dim d, d2, d3, d4, d5, d6, d7, d8, d9, jj, jjj, mm, at11 As String

Dim Dosya As String = "C:\Users\Erol\Desktop\data-A\data0.txt"

DosyaKaydet = IO.File.CreateText("C:\veri0.txt")

    If (IO.File.Exists(Dosya) = True) Then

        DosyaOku = IO.File.OpenText(Dosya)
```

```
Try

    Dim s As String = DosyaOku.ReadLine()

    s1 = 1

    s6 = 1

    s2 = 1

    s3 = 1

    s4 = 1

    s5 = 1

    s7 = 1

    s8 = 1

    s9 = 1

    s10 = 1
    s11 = 1

    denetleme = 0

    denetleme2 = 0

    s12 = 1

    s13 = 1
```

```
s14 = 1

s15 = 1

s16 = 1

s17 = 1

s18 = 1

jj = 0

jjj = 0

mm = 0

hj = 1

hjj = 1

While (s <> "")

   s = DosyaOku.ReadLine()

   If Mid(s, 1, 10) = "timesteps " And alltime = "" Then

      alltime = " = " & s.Split.GetValue(1) & ";"
      alltimes = Val(s.Split.GetValue(1))

      DosyaKaydet.WriteLine("alltime" & alltime)

   End If
```

```
If Mid(s, 1, 6) = "weeks " And week = "" Then

   week = " = " & s.Split.GetValue(1) & ";"

   DosyaKaydet.WriteLine("week" & week)

End If

If Mid(s, 1, 10) = "campaigns " And campaign = "" Then

   campaign = " = " & s.Split.GetValue(1) & ";"

   nbcamp = Val(s.Split.GetValue(1))

   DosyaKaydet.WriteLine("campaign" & campaign)

End If

If Mid(s, 1, 9) = "scenario " And scenario = "" Then

   scenario = " = " & s.Split.GetValue(1) & ";"

   nbscn = Val(s.Split.GetValue(1))

   DosyaKaydet.WriteLine("scenario" & scenario)

End If
If Mid(s, 1, 8) = "epsilon " And epsilon = "" Then

   epsilon = " = " & s.Split.GetValue(1) & ";"

   DosyaKaydet.WriteLine("epsilon" & epsilon)
```

```
End If

If Mid(s, 1, 12) = "powerplant1 " And type1 = "" Then

   type1 = " = " & s.Split.GetValue(1) & ";"

   nbtype1 = Val(s.Split.GetValue(1))

   DosyaKaydet.WriteLine("type1" & type1)

End If

If Mid(s, 1, 12) = "powerplant2 " And type2 = "" Then

   type2 = " = " & s.Split.GetValue(1) & ";"

   nbtype2 = Val(s.Split.GetValue(1))

   DosyaKaydet.WriteLine("type2" & type2)

End If
If Mid(s, 1, 10) = "durations " And d = "" And denetleme2 = 0 Then

   d2 = ""

   For xx = 1 To alltimes – 1

      d2 = d2 + s.Split.GetValue(xx) & ","

   Next

   d = " = [" & d2 & s.Split.GetValue(alltimes) & "];"
```

```
            DosyaKaydet.WriteLine("D" & d)

            denetleme2 = 1

ElseIf Mid(s, 1, 10) = "durations " And s18 <= nbtype2 And denetleme2 = 1 Then

            Dim ddmaq As String

            Dim ddmaq2 As String

            Dim gg As Integer

            For gg = 1 To nbcamp

               If gg = nbcamp Then

                  ddmaq = "[" & ddmaq + s.Split.GetValue(gg) & "]"

               Else

                  ddmaq = ddmaq + s.Split.GetValue(gg) & ","

               End If

            Next

            ddmaq2 = ddmaq2 + ddmaq
            ddmaq = ""

            If s18 = nbtype2 Then

               DosyaKaydet.WriteLine("DA = [" & ddmaq2 & "];")
```

```
    End If

    gg = 1

    s18 = s18 + 1

End If

If Mid(s, 1, 7) = "demand " And s1 <= nbscn Then

    d3 = ""

    s1 = s1 + 1

    For xx = 1 To alltimes – 1

        d3 = d3 + s.Split.GetValue(xx) & ","

    Next
    demand = demand + "[" & d3 & s.Split.GetValue(alltimes) & "]"

    If nbscn < s1 Then

        DosyaKaydet.WriteLine("DEM = [" & demand & "];")

        s1 = 1

    End If

End If

If Mid(s, 1, 5) = "pmin " And s1 <= nbscn And s6 <= nbtype1 Then
```

```
d4 = ""

Dim xx As Integer

For xx = 1 To alltimes – 1

   d4 = d4 + s.Split.GetValue(xx) & ","

Next

pmin = pmin + "[" & d4 & s.Split.GetValue(alltimes) & "]"

If s1 = nbscn Then

   pmin2 = pmin2 + "[" & pmin & "]"

   d4 = ""

End If
If s6 = nbtype1 And s1 = nbscn Then

   DosyaKaydet.WriteLine("PMIN = [" & pmin2 & "];")

   s1 = 1

   s6 = 1

End If

If s1 = nbscn Then

   s1 = 1
```

```
            s6 = s6 + 1

        Else

            s1 = s1 + 1

        End If

        xx = 0

    End If

    If Mid(s, 1, 5) = "pmax " And s2 <= nbscn And s3 <= nbtype1 And
denetleme = 0 Then

        d5 = ""

        Dim xx As Integer

        For xx = 1 To alltimes – 1

            d5 = d5 + s.Split.GetValue(xx) & ","

        Next

        pmax = pmax + "[" & d5 & s.Split.GetValue(alltimes) & "]"

        If s2 = nbscn Then

            pmax2 = pmax2 + "[" & pmax & "]"

            d5 = ""
```

```
        End If

        If s3 = nbtype1 And s2 = nbscn Then

            DosyaKaydet.WriteLine("PMAX = [" & pmax2 & "];")

            denetleme = 1

        End If

        If s2 = nbscn Then

            s2 = 1

            s3 = s3 + 1

        Else

            s2 = s2 + 1

        End If

        xx = 0

    ElseIf Mid(s, 1, 5) = "pmax " And s11 <= nbtype2 And denetleme = 1 Then

        Dim pmaxx As String
        d9 = ""

        Dim xx As Integer

        For xx = 1 To alltimes – 1
```

```
    d9 = d9 + s.Split.GetValue(xx) & ","
  Next
  pmaxx = pmaxx + "[" & d9 & s.Split.GetValue(alltimes) & "]"
  If s11 = nbtype2 Then
    DosyaKaydet.WriteLine("PMAX2 = [" & pmaxx & "];")
  End If
  s11 = s11 + 1
  xx = 0
End If
If Mid(s, 1, 5) = "cost " And s4 <= nbscn And s5 <= nbtype1 Then
  d6 = ""
  Dim xx As Integer
  For xx = 1 To alltimes – 1
    d6 = d6 + s.Split.GetValue(xx) & ","
  Next
  c1 = c1 + "[" & d6 & s.Split.GetValue(alltimes) & "]"
  If s4 = nbscn Then
```

```
        c12 = c12 + "[" & c1 & "]"

        d2 = ""

    End If

    If s5 = nbtype1 And s4 = nbscn Then

        DosyaKaydet.WriteLine("C1 = [" & c12 & "];")

        d6 = ""

    End If

    If s4 = nbscn Then

        s4 = 1
        s5 = s5 + 1

    Else

        s4 = s4 + 1

    End If

    xx = 0

End If

If Mid(s, 1, 6) = "stock " And s7 <= nbtype2 Then

    If nbtype2 = s7 Then
```

```
DosyaKaydet.WriteLine("XI = [" & d7 & s.Split.GetValue(1) & "];")

d7 = ""

Else

d7 = d7 + s.Split.GetValue(1) & ","

End If

s7 = s7 + 1

End If

If Mid(s, 1, 11) = "fuel_price " And s8 <= nbtype2 Then

If nbtype2 = s8 Then

DosyaKaydet.WriteLine("C2end = [" & d8 & s.Split.GetValue(1) & "];")

d8 = ""

Else

d8 = d8 + s.Split.GetValue(1) & ","

End If

s8 = s8 + 1

End If

If Mid(s, 1, 15) = "refueling_cost " And s9 <= nbtype2 Then
```

```
    Dim refuel As String
    Dim refuel2 As String
    For gg = 1 To nbcamp
      If gg = nbcamp Then
        refuel = "[" & refuel + s.Split.GetValue(gg) & "]"
      Else
        refuel = refuel + s.Split.GetValue(gg) & ","
      End If
    Next
    refuel2 = refuel2 + refuel
    refuel = ""
    If s9 = nbtype2 Then
      DosyaKaydet.WriteLine("C2 = [" & refuel2 & "];")
    End If
    s9 = s9 + 1
  End If
```

```
                If (Mid(s, 1, 18) = "current_campaign_s" Or Mid(s, 1, 16) =
"stock_threshold ") And s10 <= nbtype2 * nbcamp Then

                    Dim treshold As String

                    Dim streshold As String

                    Dim current As String

                    Dim hh As Integer

                    If Mid(s, 1, 18) = "current_campaign_s" Then

                       treshold = Mid(s, 34, 7) & ","

                    Else

                       For hh = 1 To nbcamp

                          If hh = nbcamp Then

                             streshold = streshold + s.Split.GetValue(hh)

                             current = current + "[" & treshold & streshold & "]"

                             streshold = ""

                             treshold = ""

                          Else

                             streshold = streshold + s.Split.GetValue(hh) & ","
```

```
            End If

        Next

    End If

    hh = 1

    If s10 = nbtype2 * nbcamp Then

        DosyaKaydet.WriteLine("BO = [" & current & "];")

    End If

    s10 = s10 + 1
End If

If Mid(s, 1, 11) = "min_refuel " And s12 <= nbtype2 Then

    Dim rmin As String

    Dim rmin2 As String

    Dim gg As Integer

    For gg = 1 To nbcamp

        If gg = nbcamp Then

            rmin = "[" & rmin + s.Split.GetValue(gg) & "]"

        Else
```

```
            rmin = rmin + s.Split.GetValue(gg) & ","

        End If

    Next

    rmin2 = rmin2 + rmin

    rmin = ""

    If s12 = nbtype2 Then

        DosyaKaydet.WriteLine("RMIN = [" & rmin2 & "];")

    End If

    gg = 1

    s12 = s12 + 1

End If

If Mid(s, 1, 11) = "min_refuel " And s12 <= nbtype2 Then

    Dim rmin As String

    Dim rmin2 As String

    Dim gg As Integer

    For gg = 1 To nbcamp

        If gg = nbcamp Then
```

```
            rmin = "[" & rmin + s.Split.GetValue(gg) & "]"
        Else
            rmin = rmin + s.Split.GetValue(gg) & ","
        End If
    Next
    rmin2 = rmin2 + rmin
    rmin = ""
    If s12 = nbtype2 Then
        DosyaKaydet.WriteLine("RMIN = [" & rmin2 & "];")
    End If
    gg = 1
    s12 = s12 + 1
End If
If Mid(s, 1, 11) = "max_refuel " And s13 <= nbtype2 Then
    Dim rmax As String
    Dim rmax2 As String
    Dim gg As Integer
```

```
    For gg = 1 To nbcamp
        If gg = nbcamp Then
            rmax = "[" & rmax + s.Split.GetValue(gg) & "]"
        Else
            rmax = rmax + s.Split.GetValue(gg) & ","
        End If
    Next
    rmax2 = rmax2 + rmax
    rmax = ""
    If s13 = nbtype2 Then
        DosyaKaydet.WriteLine("RMAX = [" & rmax2 & "];")
    End If
    gg = 1
    s13 = s13 + 1
End If
If Mid(s, 1, 13) = "refuel_ratio " And s14 <= nbtype2 Then
    Dim q As String
```

```
    Dim q2 As String

    Dim gg As Integer

    For gg = 1 To nbcamp

       If gg = nbcamp Then

          q = "[" & q + s.Split.GetValue(gg) & "]"

       Else

          q = q + s.Split.GetValue(gg) & ","

       End If

    Next

    q2 = q2 + q

    q = ""

    If s14 = nbtype2 Then

       DosyaKaydet.WriteLine("Q = [" & q2 & "];")

    End If

    gg = 1

    s14 = s14 + 1

End If
```

```
If Mid(s, 1, 16) = "max_stock_before" And s15 <= nbtype2 Then
   Dim mq As String
   Dim mq2 As String
   Dim gg As Integer
   For gg = 1 To nbcamp
      If gg = nbcamp Then
         mq = "[" & mq + s.Split.GetValue(gg) & "]"
      Else
         mq = mq + s.Split.GetValue(gg) & ","
      End If
   Next
   mq2 = mq2 + mq
   mq = ""
   If s15 = nbtype2 Then
      DosyaKaydet.WriteLine("AMAX = [" & mq2 & "];")
   End If
   gg = 1
```

```
    s15 = s15 + 1

End If

If Mid(s, 1, 15) = "max_stock_after" And s16 <= nbtype2 Then

    Dim maq As String

    Dim maq2 As String

    Dim gg As Integer

    For gg = 1 To nbcamp

        If gg = nbcamp Then

            maq = "[" & maq + s.Split.GetValue(gg) & "]"

        Else

            maq = maq + s.Split.GetValue(gg) & ","

        End If

    Next

    maq2 = maq2 + maq

    maq = ""

    If s16 = nbtype2 Then

        DosyaKaydet.WriteLine("SMAX = [" & maq2 & "];")
```

```
                End If

                gg = 1

                s16 = s16 + 1

            End If

            If (Mid(s, 1, 18) = "current_campaign_m" Or Mid(s, 1, 12) =
"max_modulus ") And s17 <= nbtype2 * nbcamp Then

                Dim modul As String

                Dim current As String

                Dim moducur As String

                Dim hh As Integer

                If Mid(s, 1, 18) = "current_campaign_m" Then

                   modul = Mid(s, 30, 6) & ","

                Else

                   For hh = 1 To nbcamp

                      If hh = nbcamp Then

                         current = current + s.Split.GetValue(hh)

                         moducur = moducur + "[" & modul & current & "]"
```

```
            current = ""
            modul = ""
          Else
            current = current + s.Split.GetValue(hh) & ","
          End If
        Next
      End If
      hh = 1
      If s17 = nbtype2 * nbcamp Then
        DosyaKaydet.WriteLine("MMAX = [" & moducur & "];")
      End If
      s17 = s17 + 1
    End If
    If Mid(s, 1, 15) = "profile_points " Then
      points = s.Split.GetValue(1)
    End If
    If Mid(s, 1, 17) = "decrease_profile " Then
```

```
If hjj = nbtype2 Then

If hj = nbcamp + 1 Then

firsta = "[" & firsta + "<" & s.Split.GetValue(points * 2 - 1) & "," & s.Split.GetValue(points * 2) & ">]"

For gh = points - 2 To 2 Step -1

If gh = 2 Then

nrmal = "[{" & nrmal + "<" & s.Split.GetValue(gh * 2 - 1) & "," & CStr((Val(s.Split.GetValue(gh * 2)) - Val(s.Split.GetValue(gh * 2 - 2))) / (Val(s.Split.GetValue(gh * 2 - 1)) - Val(s.Split.GetValue(gh * 2 - 3)))) & ">}]"

Else

nrmal = nrmal + "<" & s.Split.GetValue(gh * 2 - 1) & "," & CStr((Val(s.Split.GetValue(gh * 2)) - Val(s.Split.GetValue(gh * 2 - 2))) / (Val(s.Split.GetValue(gh * 2 - 1)) - Val(s.Split.GetValue(gh * 2 - 3)))) & ">,"

End If

Next

att3 = att3 + nrmal

att2 = att2 + firsta

DosyaKaydet.WriteLine("PBfirst = [" & att2 & "];")

DosyaKaydet.WriteLine("PB = [" & att3 & "];")
```

```
                Else

                    firsta = firsta + "<" & s.Split.GetValue(points * 2 - 1) & "," &
s.Split.GetValue(points * 2) & ">,"

                    hj = hj + 1

                End If

            Else

                If hj = nbcamp + 1 Then

                    firsta = "[" & firsta + "<" & s.Split.GetValue(points * 2 - 1) & ","
& s.Split.GetValue(points * 2) & ">],"

                    For gh = points - 1 To 2 Step -1

                        If gh = 2 Then

                            nrmal = "[{" & nrmal + "<" & s.Split.GetValue(gh * 2 - 1)
& "," & CStr((Val(s.Split.GetValue(gh * 2)) - Val(s.Split.GetValue(gh * 2 - 2))) /
(Val(s.Split.GetValue(gh * 2 - 1)) - Val(s.Split.GetValue(gh * 2 - 3)))) & ">}],"
                        Else

                            nrmal = nrmal + "<" & s.Split.GetValue(gh * 2 - 1) & "," &
CStr((Val(s.Split.GetValue(gh * 2)) - Val(s.Split.GetValue(gh * 2 - 2))) /
(Val(s.Split.GetValue(gh * 2 - 1)) - Val(s.Split.GetValue(gh * 2 - 3)))) & ">,"

                        End If

                    Next
```

```
                att3 = att3 + nrmal

                att2 = att2 + firsta

                firsta = ""

                nrmal = ""

                hjj = hjj + 1

                hj = 1

            Else
                firsta = firsta + "<" & s.Split.GetValue(points * 2 - 1) & "," &
s.Split.GetValue(points * 2) & ">,"

                For gh = points - 1 To 2 Step -1

                    If gh = 2 Then

                        nrmal = "{" & nrmal + "<" & s.Split.GetValue(gh * 2 - 1) &
"," & CStr((Val(s.Split.GetValue(gh * 2)) - Val(s.Split.GetValue(gh * 2 - 2))) /
(Val(s.Split.GetValue(gh * 2 - 1)) - Val(s.Split.GetValue(gh * 2 - 3)))) & ">},"

                    Else

                        nrmal = nrmal + "<" & s.Split.GetValue(gh * 2 - 1) & "," &
CStr((Val(s.Split.GetValue(gh * 2)) - Val(s.Split.GetValue(gh * 2 - 2))) /
(Val(s.Split.GetValue(gh * 2 - 1)) - Val(s.Split.GetValue(gh * 2 - 3)))) & ">,"
                    End If

                Next
```

```
            hj = hj + 1

        End If

    End If

End If

If Mid(s, 1, 13) = "constraint13 " Then

    ct13 = Val(s.Split.GetValue(1))

End If

If Mid(s, 1, 7) = "type 13" Then

    Dim nbcampaign, nbpower As Integer

    Dim east, least, eastson, leastson, at1, at2 As String
    jj = jj + 1

    jjj = jjj + 1

    s = DosyaOku.ReadLine()

    s = DosyaOku.ReadLine()

    If Mid(s, 1, 11) = "powerplant " Then

        nbpower = Val(s.Split.GetValue(1)) + 1

    End If
```

```
s = DosyaOku.ReadLine()

If Mid(s, 1, 9) = "campaign " Then

    nbcampaign = Val(s.Split.GetValue(1)) + 1

End If

s = DosyaOku.ReadLine()

If Mid(s, 1, 19) = "earliest_stop_time " Then
    east = s.Split.GetValue(1)

End If

s = DosyaOku.ReadLine()

If Mid(s, 1, 17) = "latest_stop_time " Then
    least = s.Split.GetValue(1)

End If

If jjj = nbcamp * nbtype2 Then

    If jj = nbcamp Then

  eastson = "{" & eastson + "<" & nbcampaign & "," & east & ">}"

  leastson = "{" & leastson + "<" & nbcampaign & "," & least & ">}"

       at1 = "[" & at1 + eastson & "];"

       at2 = "[" & at2 + leastson & "];"
```

```
        DosyaKaydet.WriteLine("east = " & at1)

        DosyaKaydet.WriteLine("Ista = " & at2)

        jj = 0

    ElseIf jj < nbcamp Then

        eastson = eastson + "<" & nbcampaign & "," & east & ">, "

        leastson = leastson + "<" & nbcampaign & "," & least & ">, "

    End If

ElseIf jjj < nbcamp * nbtype2 Then

    If jj = nbcamp Then

 eastson = "{" & eastson + "<" & nbcampaign & "," & east & ">}, "

 leastson = "{" & leastson + "<" & nbcampaign & "," & least & ">}, "

        at1 = at1 + eastson

        at2 = at2 + leastson

        eastson = ""

        leastson = ""

        jj = 0

    ElseIf jj < nbcamp Then
```

```
            eastson = eastson + "<" & nbcampaign & "," & east & ">, "

            leastson = leastson + "<" & nbcampaign & "," & least & ">, "

        End If

    End If

End If

If Mid(s, 1, 13) = "constraint14 " Then

    ct14 = Val(s.Split.GetValue(1))

End If

If Mid(s, 1, 7) = "type 14" Then

    mm = mm + 1

    Dim ik As Integer

    Dim set14 As String

    If mm = ct14 Then

        DosyaKaydet.WriteLine("m14 = " & mm - 1 & ";")

    End If

    s = DosyaOku.ReadLine()

    s = DosyaOku.ReadLine()
```

```
ik = s.Split.Count

If mm = ct14 Then

   For n = 1 To ik – 1

      If n = ik - 1 Then

        set14 = "{" & set14 + CStr(Val(s.Split.GetValue(n)) + 1) & "}"

         at11 = at11 + set14

         set14 = ""

         DosyaKaydet.WriteLine("A14 = [" & at11 & "];")

      Else

         set14 = set14 + CStr(Val(s.Split.GetValue(n)) + 1) & ","

      End If

   Next

Else

   For n = 1 To ik – 1

      If n = ik - 1 Then

        set14 = "{" & set14 + CStr(Val(s.Split.GetValue(n)) + 1) & "},"

         at11 = at11 + set14
```

```
                    set14 = ""

                Else

                    set14 = set14 + CStr(Val(s.Split.GetValue(n)) + 1) & ","

                End If

            Next

        End If

    End If

End While

Finally

    DosyaOku.Close() 'Dosyayı kapatıyor.

    DosyaKaydet.Close()

End Try

End If

Close()
End Sub

End Class
```

EK C

```
alltime=623;

week=89;

campaign=2;

scenario=2;

epsilon=0.01;

type1=1;

type2=2;

D=[24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,
24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,2
4,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24
,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,
24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,2
4,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24
,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,
24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,2
4,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24
,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,
24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,2
4,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24
,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,
24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,2
4,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24
,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,
24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,2
4,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24
,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,
24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,2
4,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24
,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,
24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24,24];


DEM=[[51318.2,49847,51838.9,51954.7,51179.6,50547.5,51043,51271.8,49524.1,5
4698.3,54744.6,55471.9,54682.6,55446.6,52487.8,50128.8,53525.3,55031,55385.5,5
5227.5,55521.9,55325.9,53172.9,56015.8,56879.6,59438.4,58819.9,58532.6,55484.2
,53291.1,58617.7,58388.8,60185.9,58212.2,59034.3,56331.3,53832.7,59471.5,58397
.3,59757.9,61104.8,62377.8,60361.7,62876,61598.1,62856.9,62627.7,63725.6,62756
```

.3,61584.9,59219.1,62701,62165.1,65653.7,66046.7,68562.8,63035.7,59196.1,65293
.8,64642.1,63605.1,63094.1,56830.2,61267.1,59103.3,64456.4,65815.9,64827.5,642
13.9,63627.9,60596.9,59565.7,67726.9,68411.5,71135.6,69739,71950,68919.7,6688
2.1,72812.3,73088,72989.5,72559.5,75673.6,68766,67011.6,74393.4,75998.3,76563,
73832.3,74450.1,70083.5,66858.5,72274,76364.4,75657.3,74114.6,76709.5,69843,6
5853.9,69789.4,66676.5,69345.3,74101.8,74451.4,69795.3,67901,72194.5,72349.2,7
1119.3,71693.2,69638.5,64309.6,60128.3,64481.8,70243.6,71726.9,69604.4,68905,6
3741.2,63193.7,72898.4,74093.4,75118.3,73998.4,75951.4,75541,73859.8,81225.7,8
1814.5,80937.6,79177.3,78882.3,72323.5,69058.8,73035.6,71991.8,72592.9,70811.7
,69994.6,64141.3,61705.2,70024.4,69049.9,67522.5,65964.3,66091.5,60126.6,54993
,61054.9,62523.4,64226.3,64039.4,64452.2,58672.5,56861.4,63262.3,61216.2,61001
.4,60999.2,60999.2,58215.2,56327.5,58856.9,59165.7,59271.3,59453.2,59252.2,579
05.8,55976,58544.3,58645,58683.8,57995.2,58356.7,55891.5,56118.3,59167.1,5796
0.6,57335.8,57722.1,58400,57492.4,56043.6,59932.9,57082.1,58982.7,57335.5,5770
0.4,53227,53433.5,58191.6,57156.7,60304.3,57631.2,56828.2,57007.2,53817.8,6004
8.2,59425.9,59119.2,59737.2,60474.6,53403.5,53000.8,55211.1,54166.2,54715.3,56
698.1,56244.3,53594.1,51774.3,56316.9,56926.7,56951.7,56738,56465.4,50156,485
68.2,48970.8,55113.8,57202.4,58946.4,55839.1,54749.8,51959.8,56790.7,57761,581
07.4,56585.4,58368.6,51202.4,45406.7,45885.5,53296.9,56322.5,56369.4,55977.7,5
1844.7,49102.8,50658.8,53912.5,55143.9,51855.8,52876.9,50765.2,48363.2,52984.4
,54589.5,54232.6,55860.3,54089.7,50301.8,47609.8,52676.3,52349.6,50218.7,44695
.1,47283.4,46755.8,46451.2,53556.5,54128.8,53140.7,50930.6,48972,45379,42293.2
,48982.6,49282.5,51217.8,51656.4,51192.8,45024.5,43990.1,49630.7,47353.2,48366
.3,49619.3,49535.7,49142,46887.3,53397.2,53036.1,53658.7,50699.4,53739.8,48076
.1,45933,50837.3,50751.2,51750.8,50883.7,51478.9,51291,47970.7,53221.5,52433.5
,52733.1,53188.2,54473.4,47929.3,48126.6,50134.6,51712.4,51980.1,51248.1,49013
.1,51410.8,49088.5,52326.8,53607.8,54040.9,53322.9,52781.1,52079.2,49937,55051
.1,56166.8,56656.1,56625.1,56133.6,51932.4,48400.7,52753.2,51925.7,53487,52559
.1,52755.2,49563.6,46652.2,50651.6,51396,51207.6,51378.6,52015.5,49643.8,46577
.5,51089.3,48843,53026.5,53426.4,53780.7,50305,47613.2,52167.7,54435.1,54857.1
,55096,56057.4,52681.4,49342.9,54456.7,54622.5,56699,55654.1,57919.5,55618,52
796.4,55795.9,57145,57373.3,57566.9,55372.5,52298.7,50662.8,57032.1,57340.5,53
319.6,53223.5,55344,51341.2,48823,53092.6,51887.6,53081.5,53599.5,54355.5,511
06.2,49589.6,52862.1,54921.4,56120.7,54649.9,55270.3,53830,52085.7,57459,5869
1.4,56174,56215.9,56586.4,52378.9,50081.1,53951.8,54458.3,57292.5,58525.5,5866
7.8,54065.9,51611,56861.2,59331,60715.4,57988.3,58926.6,56603.4,56343.1,61204.
4,61373.9,58167.1,61141.5,60340.7,56615.1,48732.4,56168.5,58411.6,60343.9,6217
7.1,63067.3,61624.6,58519.7,62210.7,62436.4,63449.5,61875.2,61921.2,58121.3,58
794.8,63933.1,61653,60944.2,61098.1,60192.8,59231.1,58401.6,62845.7,61628.9,65
113.6,62315.9,64088.5,69239.3,66131.1,68613.5,72146.1,70437.1,69858.4,70460.2,
65825.4,63528.1,71071.4,73874.4,74764.8,77685,76299.6,71518.8,68977.8,75779,7
4283,73795.1,73576.5,74417.1,73656.2,68151.4,74666.5,72950.9,72486.1,71775.4,7
4461.5,72274.9,70418.3,71550.1,77176.8,79493.5,79763.6,76810.5,71458.3,68119.3
,65443.3,70324.2,74535.1,75545.4,72679.7,68037.2,66856,74809.6,76959.5,77478.5
,77764.4,78434.1,79044.3,78440.4,81769.6,79700.5,79519.1,79261,77738.5,73609.8
,71914.4,75791.3,74288.3,74695.7,75635.1,74610,66418.6,64137.6,69631.3,71671.5
,73557.2,70155.1,69775.2,62867.1,61054.4,66390.6,67348.6,66863.5,68102.8,68409
.6,65326.2,63788.1,69965.5,69189.4,70836,68633.6,68858.6,60046.2,55195.6,63733
.3,65943.9,65120.6,64282.4,66154.8,59422.1,56264,60737.7,59172.7,59625.7,60817

.2,65721,64672.3,61221.7,66846.4,62922.4,62583.7,64705.3,64174.3,62522.1,62133
.7,61883.7,62611.2,63798.7,62340.1,63200,62317,62218.8,68762.8,67117.2,65899.5
,65476.3,69215.4,61713.1,58776.7,60998,61836.4,63485,64563.3,62835.4,59060.4,5
4948.3,58697.8,54890.3,56366.8,56136.6,59279.7,53593.1,51810.1,53029,56842.8,5
7294.7,55526.6,55538.8,54794.1,52153.4,56165.4,57546.7,57809.2,58240.6,57809.9
,56220.9,52168,59105.5,60185,60207.4,60403,60418.4,55863.5,51335.6,52938.8,46
601.6,52293.3,55461.6,53860.9,51365.8,48971.3,55494.8,52006.2,57155.9,57607.2,
56380.4,50369.2,44768.7,53083.1,55224.5,56217.5,46663.6,50587,48778.2,42959.4,
51530.1,51871.7,52337,52591.1,51268.6]
[51629.4,48625.5,52573.8,52329.5,53132.4,53063.6,52945.8,53158.3,49950.7,5295
3,53364.8,53408.1,53789.2,54891.2,52393.6,48629.5,53998.1,53111.9,51986.7,5184
4.1,52312.1,50364.2,47869.3,54863.6,55742.1,56156.1,54624.9,54545.8,51128.5,49
580.3,54114.8,55184.7,55074.2,54272.4,55084.9,55287.3,52995.2,56609.7,56651.3,
57977.8,58832.6,58986.8,54305.2,55311.6,60255.9,60094.3,57513.1,58295.5,59488.
1,60325.1,58523.1,62014.5,62473,65737.3,66921.2,67447.3,63716.7,58097.5,64533.
3,65219.5,66908.2,64266.4,54138.4,54447.8,54118.6,64609.8,64824.7,64563.7,6214
8.9,60980.4,57742,57051.4,63706.9,61912.8,62656.6,66941.6,65062.5,61433,60685.
8,69440.9,70375.9,69134,74470.9,75317.2,73818.4,69871.7,70889,72910.5,70518.7,
72978.2,73817.2,65065.3,61413.3,69271.8,71910.5,69988.5,68793.5,70301.8,65809.
2,62567.9,67449,69801.6,70668.1,70645.7,68485.8,62024.6,58522.1,67006.2,69997.
7,73156.1,75401.4,75161.7,70180.6,67641.7,78012.6,81614,83214.5,80831.7,82784,
79374,76747.7,82584.1,80293.8,82342.1,80299.1,81220.6,76731.5,76472.7,78175,7
8953.7,79228.8,75095.8,72211.5,66542.5,64344.2,71748.8,73684.3,73273.8,71139.3
,72077.1,67291.1,66087.6,73292.1,73602.2,71612,73600.6,76563.7,72303.2,71744.7
,72754.6,73802.7,70767.9,71281.2,69819.5,62942.5,60885.3,67328.5,69772.1,69959
.7,69294.9,67016.6,56278.5,56734.1,66592.4,68480.7,65954,65366.1,66137.1,64181
.9,62043.1,67745.1,63796.5,63637.8,63667.1,64153,60139.5,57821.1,63021.3,61859
.4,62389.9,62988.2,60111.4,60360.4,58102,62233.8,61551.5,62120,59551.2,64052.1
,60808.6,60342.9,59880.1,59948.9,62654.6,61419.6,60683.4,61786.5,57581.9,60823
.1,61039.4,60782.2,65072.9,63755.9,63471.6,62139.6,61199.2,63672.6,61518.4,626
29.4,62613.3,60273.3,59211,60537.6,61290.8,60287.9,58463.3,58641.9,55983,5352
9.2,53420.7,56521.5,58777.1,59029.8,59244.4,56094,53084.2,58414.6,59416.6,5831
6.5,58729.4,58555.5,54712.3,50455.8,51016.1,58195.1,59767.1,57883.3,60200,5612
9.5,54597.8,55375.7,56989.1,55244.9,55152.5,57472.6,55362.3,52288.3,56611.2,55
577.7,56548,57328.9,57007.2,54245,50890,55865.5,56521.1,55344.5,50041.3,49809
.2,49850.8,48016.5,54852.4,54733.9,54608.4,54296.8,53305,49974.7,46047.9,52364
.9,50731.9,50559.2,53116.8,53047.6,49581.1,46606.9,52577.9,52233.1,54143.4,540
64.4,53951.2,51981.1,49584.8,54674.8,55082.3,55314,54085.2,56657.9,52794.8,508
12.8,53950.9,53474.6,54856.8,53365.7,52953.2,51169.3,48665.4,53727.5,53829.1,5
2650.4,52715,53912.1,52677.5,50879,55025.2,53668.6,53106.9,54037.5,48500.4,51
621.2,49322.3,55067.6,55621.8,56538.9,57113.7,56744.1,53171.1,50247.6,56388.8,
57917.4,58272.9,57928.9,57101.8,53124.4,48932.3,52628.8,52313.6,51424.9,50863.
1,50492.1,47606.3,44107.9,47362.9,45938,47747.5,46110.1,49078.8,47942.4,44573.
8,47538.7,46099.6,49708.2,50306.5,51190.6,48786.9,45370.1,51965.8,52009.9,5209
8.4,52720.1,52840.9,50224.7,47228.8,54005.1,55412.6,54875.6,53379.1,55089.3,51
943.5,50363.6,55354.7,55125.7,55373.1,56780.4,55284,51934.8,49710.1,54779.7,55
477.4,55380,57574.1,56360.6,45315.5,42973.4,50856.8,52721.6,54452.8,53908.3,54
012.4,49200.1,47869.2,53675.6,54383.3,54725.7,54398.2,52109.9,50616.5,53058.4,
58835.2,56396.5,58806,57050.7,59452.7,55854.1,53725.2,58703.7,59615.8,59918.8,

60441.7,61307,59079.3,54288.1,60349.1,60381.1,60320.4,60477.8,60497.8,56698.5,
55247.1,62794.3,62017.2,62886.9,64725.5,64550.2,63816.3,62632.3,64963.1,67940.
8,66015.9,70334.7,70285.1,69820.5,66601.1,70379.6,69564.4,68410.2,68959.2,6895
3.2,57575.5,57277.3,67336.3,69148.1,70030.6,70684.6,68143.5,63277.8,61132.8,68
197.3,68411.7,67906.7,67817.5,68245.9,63698.2,62282.4,68353.6,71278.2,72103.9,
71574.4,71718.7,65558.9,66599,73272.9,72665.7,72158.2,69328,71084.5,63925.4,5
8277.2,65906.6,65470.4,66355.6,68230.2,69017.8,64823.4,62393.4,71266.9,68262.4
,69971,70387,70530.8,67839,61540.5,60603.5,68674.9,71111.2,73921.5,76172.8,73
681.9,68789.6,71042.4,79812.3,83028.9,84575.3,84580.9,80907.3,80139.6,82679.9,
83754.6,83574.8,82639.6,81121.6,82344.3,79316.3,82103.6,82281.7,83836.8,85014.
7,80898.9,72575.9,68236.8,74791.6,75710.3,71856.8,72689.4,71717.9,67431.9,6603
0.2,75387.8,75672,74995.1,78470.3,79833.3,76012.3,74718.4,77449.6,78706.4,7557
8.8,75170.7,75408.7,68104.2,64729.7,72580.3,72216,73462.5,74485.6,72828.8,6602
4,63274.6,67789.4,68099.2,69476,71151.5,73526.1,68461.6,66650.6,69311,66688.8,
68918.2,67013.2,65804.3,58803.5,57132.1,65177.6,64162.2,61884,64361.1,66469.2,
62341.4,58723.6,62956.3,63108.9,64102,63497.8,65588.4,62330.2,59633.2,63442.4,
63108.2,64911,64664.7,66380.5,62317.2,59613.7,61774.7,59801.7,59887,60317.5,6
1565.6,64751.3,61418.6,64355.9,62210.7,63821.9,63500.4,63367.6,59473.6,57838.4
,57825.4,60135.6,60695.6,59894.1,58157.1,55875.7,51506.4,56384.6,56200.6,57948
.1,55303.2,55833.8,53279.3,50218.3,54623.7,56335,59482.9,59407.3,57316.8,57312
.6,52422.5,52952.5,52082.3,57587.5,57537.4,57573.7,56675,52925.2,57861.1,56650
.8,59720.2,59819.5,60511.6,55539.6,52237.7,58560,59197.3,58154.4,51829.8,51978
.2,52400.2,48534.6,55700,57692.9,58048.9,58022.6,57073.2]
];

PMIN=[[[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

```
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
]
];


PMAX=[[[100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
```

00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1000
00,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,1
00000,100000,100000,100000,100000]
[100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10

0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10000
0,100000,100000,100000,100000,100000,100000,100000,100000,100000,100000,10
0000,100000,100000,100000]
]
];

C1=[[[10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100

00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000]
[10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100

```
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1
0000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,100
00,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,
10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10
000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,10000,1000
0,10000,10000,10000]
]
];


XI=[4974480,6866640];


C2end=[20.68,20.26];
```

C2=[[20.37,19.38]
[19.96,19.02]
];

BO=[[1764000,1764000,1764000]
[1836000,1836000,1836000]
];

PMAX2=[[1335,1335,1335,1335,1335,1335,1335,1320,1320,1320,1320,1320,1320,
1320,1320,1320,1320,1320,1320,1320,1320,1350,1350,1350,1350,1350,1350,1350,1
365,1365,1365,1365,1365,1365,1365,1395,1395,1395,1395,1395,1395,1395,1395,13
95,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,139
5,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395
,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1410,1410,1410,1410,1410,
1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1
410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,14
10,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,141
0,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410
,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,
1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1
410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,14
10,1410,1410,1410,1410,1410,1410,1410,1395,1395,1395,1395,1395,1395,1395,139
5,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395
,1395,1395,1395,1395,1395,1380,1380,1380,1380,1380,1380,1380,1350,1350,1350,
1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1
350,1350,1350,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,13
65,1365,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,138
0,1380,1380,1380,1380,1380,1380,1380,1365,1365,1365,1365,1365,1365,1365,1365
,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,
1365,1365,1365,1365,1365,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1
350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1365,1365,1365,1365,13
65,1365,1365,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,138
0,1380,1365,1365,1365,1365,1365,1365,1365,1350,1350,1350,1350,1350,1350,1350
,1350,1350,1350,1350,1350,1350,1350,1365,1365,1365,1365,1365,1365,1365,1380,
1380,1380,1380,1380,1380,1380,1395,1395,1395,1395,1395,1395,1395,1395,1395,1
395,1395,1395,1395,1395,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,14
10,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1395,1395,1395,1395,139
5,1395,1395,1395,1395,1395,1395,1395,1395,1395,1410,1410,1410,1410,1410,1410
,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,
1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1
410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,14
10,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,141
0,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410
,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,
1410,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1
380,1380,1380,1380,1380,1380,1380,1365,1365,1365,1365,1365,1365,1365,1365,13
65,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,136

5,1365,1365,1365,1365,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380
,1380,1380,1380,1395,1395,1395,1395,1395,1395,1395,1380,1380,1380,1380,1380,
1380,1380]
[1335,1335,1335,1335,1335,1335,1335,1320,1320,1320,1320,1320,1320,1320,1320,
1320,1320,1320,1320,1320,1320,1350,1350,1350,1350,1350,1350,1350,1365,1365,1
365,1365,1365,1365,1365,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,13
95,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,139
5,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395
,1395,1395,1395,1395,1395,1395,1395,1395,1410,1410,1410,1410,1410,1410,1410,
1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1
410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,14
10,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,141
0,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410
,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,
1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1
410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,14
10,1410,1410,1410,1410,1410,1395,1395,1395,1395,1395,1395,1395,1395,1395,139
5,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395
,1395,1395,1395,1380,1380,1380,1380,1380,1380,1380,1350,1350,1350,1350,1350,
1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1
350,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,13
80,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,138
0,1380,1380,1380,1380,1380,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365
,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,
1365,1365,1365,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1350,1
350,1350,1350,1350,1350,1350,1350,1350,1350,1365,1365,1365,1365,1365,1365,13
65,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,136
5,1365,1365,1365,1365,1365,1365,1350,1350,1350,1350,1350,1350,1350,1350,1350
,1350,1350,1350,1350,1350,1365,1365,1365,1365,1365,1365,1365,1380,1380,1380,
1380,1380,1380,1380,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1
395,1395,1395,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,14
10,1410,1410,1410,1410,1410,1410,1410,1410,1395,1395,1395,1395,1395,1395,139
5,1395,1395,1395,1395,1395,1395,1395,1410,1410,1410,1410,1410,1410,1410,1410
,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,
1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1
410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,14
10,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,141
0,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410
,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1410,1395,
1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1395,1380,1
380,1380,1380,1380,1380,1380,1365,1365,1365,1365,1365,1365,1365,1365,1365,13
65,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,1365,136
5,1365,1365,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380,1380
,1380,1395,1395,1395,1395,1395,1395,1395,1380,1380,1380,1380,1380,1380,1380]
];

RMIN=[[9102240,9102240]
[6462720,6462720]

```
];


RMAX=[[9102240,9102240]
[12484800,12484800]
];


Q=[[4,4]
[4,4]
];


AMAX=[[3175200,3175200]
[3304800,3304800]
];


SMAX=[[14112000,14112000]
[14688000,14688000]
];


MMAX=[[176400,176400,176400]
[183600,183600,183600]
];


DA=[[5,8]
[9,6]
];


PBfirst=[[<0,0.74> <0,0.74> <0,0.74> ]
[<0,0.74> <0,0.74> <0,0.74> ]
];


PB=[[{<352800,1.41723e-007>,<705600,1.41723e-007>,<1058400,1.70068e-
007>,<1411200,1.41723e-007>,<1764000,1.41723e-007>}
{<352800,1.41723e-007>,<705600,1.41723e-007>,<1058400,1.70068e-
007>,<1411200,1.41723e-007>,<1764000,1.41723e-007>}
{<352800,1.41723e-007>,<705600,1.41723e-007>,<1058400,1.70068e-
007>,<1411200,1.41723e-007>,<1764000,1.41723e-007>}
]
[{<367200,1.36166e-007>,<734400,1.36165e-007>,<1101600,1.63399e-
007>,<1468800,1.36166e-007>,<1836000,1.36166e-007>}
{<367200,1.36166e-007>,<734400,1.36165e-007>,<1101600,1.63399e-
007>,<1468800,1.36166e-007>,<1836000,1.36166e-007>}
```

{<367200,1.36166e-007>,<734400,1.36165e-007>,<1101600,1.63399e-007>,<1468800,1.36166e-007>,<1836000,1.36166e-007>}
]
];

east=[{<1,18>, <2,56>},
{ <1,24>, <2,79>}];

Ista=[{<1,26>, <2,64>},
{<1,32>, <2,87>}];

m14=0;

Sem14=[6];

A14=[{0,1}];

ÖZGEÇMİŞ

Erol Aslan, 01.02.1985 de Bursa'da doğdu. İlk, orta ve lise eğitimini İstanbul'da tamamladı. 2003 yılında Ümraniye (yda) Lisesinden mezun oldu. 2004 yılında başladığı Sakarya Üniversitesi Endüstri Mühendisliği Bölümünü 3 yılda tamamlayarak 2007 yılında mezun oldu. 2007 yılından beri Cedetaş Elektromekanik Sanayi ve Ticaret A.Ş. de Üretim Planlama Sorumlusu olarak çalışmaya devam etmektedir.