TP SORUNLARININ ÇÖZÜMLERİ

(*Bu notlar* ***Doç.Dr. Şule Önsel*** *tarafıdan hazırlanmıştır*)

TP problemlerinin çözümü için başlıca iki yaklaşım vardır.

İlk geliştirilen yöntem ***kesme düzlemleri*** (cutting planes) olarak isimlendirilir. Tamsayılığı sağlamak için kısıtlar eklemeye dayanır.

1980'lerden beri kullanılan ve daha etkin olarak nitelendirilen yöntem ise ***dal sınır*** (branch and bound) olarak isimlendirilir. Sorunu alt sorunlara bölen bir ağaç arama yaklaşımıdır.

Aslında, tüm yaklaşımlar tekrarlı bir şekilde DP'ler çözmeyi içerir. DP çözümü için g*enel amaçlı* (çözülen DP'den bağımsız herhangi bir DP'yi çözebilecek yapıda) ve *etkin işlem yapabilen* (büyük DP'leri çözebilen) algoritmalar (örn. simpleks) vardır. TP çözümü için ise hem genel amaçlı hem de etkin işlem yapabilen benzer bir algoritma yoktur.

TP için çözüm yöntemleri iki sınıfa ayrılabilir:

Genel amaçlı (herhangi bir TP'yi çözebilecek yapıda) fakat etkin işlem yapamayan (göreli olarak daha küçük sorunları çözen); veya

Özel amaçlı (Belirli bir tipteki TP sorunu için geliştirilen) ve daha etkin işlem yapabilen.

TP çözüm yöntemleri aynı zamanda çözüme gitmeleri açısından aşağıdaki gibi de sınıflandırılabilir:

En iyi (Optimal)

Sezgisel (Heuristic)

"En iyi" algoritma matematiksel olarak en iyi çözümü bulmayı *garantiler*

Bazı durumlarda en iyi çözümle çok fazla ilgilenmeyebiliriz çünkü:

- çözmek istediğimiz sorunun büyüklüğü mevcut en iyi algoritmalarının etkin işlem yapabilme sınırlarının üstündedir (kullanabileceğimiz bilgisayar zamanı gibi)
- harcanacak zamana, paraya en iyi çözümü bulmak değmez, yaklaşık bir çözüm yeterli görülebilir

Bu gibi durumlarda sezgisel algoritmalar kullanılabilir. Bulunan çözüm olurlu bir çözümdür ve büyük bir olasılıkla iyi tasarlanmış bir yöntem kullanılıyorsa en iyi çözüme yakın kaliteli bir çözümdür.

Bu durumda çözüm algoritmaları için dört farklı kategori vardır:

Genel Amaçlı, En iyi

Sayma, Dal sınır, kesme düzlemi

Genel Amaçlı, Sezgisel (*ders kapsamında değil*)

Genel amaçlı bir en iyi algoritmasını çalıştırmak ve belirli bir süre sonunda durmak

Özel Amaçlı, En iyi (*ders kapsamında değil*)

Sınır üretmeye dayalı ağaç arama yaklaşımları

Özel amaçlı, Sezgisel (*ders kapsamında değil*)

Sınır tabanlı sezgiseller, tabu arama, genetik algoritmalar...

DP Gevşetmesi (DP ile İlişki)

Herhangi bir TP için aynı amaç fonksiyonu ve kısıtları kullanarak fakat değişkenlerin tamsayı olma gereksinimini aşağıdaki gibi değiştirerek bir DP elde edebiliriz:

"x_i = 0 veya 1" değişkeni 0 <= x_i <= 1 sürekli aralığında değerler alır

"$x_i \geq 0$ ve tamsayı" değişkeni salt x_i >= 0 işaret sınırlamasını sağlar

Değişkenler üzerindeki tüm tamsayı ve 0-1 şartlarını yoksayarak elde edilen DP, ***TP'nin DP Gevşetmesi*** (LP Relaxation of the IP) olarak isimlendirilir. Söz konusu doğrusal gevşetmeyi (DG) DP çözüm algoritmalarını kullanarak çözebiliriz.

Eğer DG'nin en iyi çözümündeki tüm değişkenler tamsayılı değerler alıyorlarsa bu durumda bulunan en iyi çözüm orijinal TP sorununun da en iyi çözümüdür (*doğal tamsayılı DP*)

DG, TP sorununa göre daha az kısıtlı (gevşek) olduğundan aşağıdaki durumlarla karşılaşılabilir:

Eğer TP enbüyükleme sorunu ise, DG'nin en iyi amaç değeri TP'ninkine eşit veya daha büyüktür.

Eğer TP enküçükleme sorunu ise, DG'nin en iyi amaç değeri TP'ninkine eşit veya daha küçüktür.

Eğer DG olurlu değilse (olurlu çözümü yoksa), TP de olurlu değildir.

Bu durumda DG'nin çözülmesinin bir bilgi vereceği açıktır: en iyi amaç değeri sınırı belirlenir ve şanslı isek en iyi amaç değerini buluruz. Fakat daha önce de gördüğümüz gibi en iyi çözümdeki değişken değerlerini tamsayı değerlere yuvarlamak genel olarak en iyi TP çözümünü vermeyebilir; hatta yeni çözüm olurlu bile olmayabilir.

Genel olarak DP tabanlı TP çözüm paketleri aşağıdaki çözüm sürecini kullanır

TP'nin belirlenmesi: amaç, kısıtlar, tamsayılı değişkenler (x_j = tamsayı ve 0 ile n arasında).

DP Gevşetmesinin yapılması: TP'deki amaç ve kısıtlara ek olarak $0<=x_j<=n$ fakat tamsayı olma şartının kaldırılması.

Dal sınır algoritması ile TP'nin en iyi çözümünün bulunması

Sayma

DP'den (değişkenlerin sürekli aralıkta değerler alabildiği (>=0)) farklı olarak TP'de (tüm değişkenler tamsayılı) her değişken sadece sonlu sayıda kesikli (tamsayı) değerler alabilir.

Tüm olası çözümleri ***sayma*** (enumerate) – her biri için amaç fonksiyon değerini hesaplama ve olurlu çözümlerden en iyisini seçme – bir çözüm yaklaşımı olabilir.

Örneğin aşağıdaki çok dönemli sermaye bütçeleme sorununu inceleyelim,

Maks $0.2\, x_1 + 0.3\, x_2 + 0.5\, x_3 + 0.1\, x_4$

Öyle ki $0.5\, x_1 + 1\, x_2 + 1.5\, x_3 + 0.1\, x_4 \leq 3.1$

$0.3\, x_1 + 0.8\, x_2 + 1.5\, x_3 + 0.4\, x_4 \leq 2.5$

$0.2\, x_1 + 0.2\, x_2 + 0.3\, x_3 + 0.1\, x_4 \leq 0.4$

$x_j = 0$ or $1 \qquad j = 1, \ldots 4$

Bu sorun için 2^4=16 olası çözüm vardır. Bunlar:

0 0 0 0 hiç proje uygulama	0 0 1 1 iki proje uygula	1 1 1 0 üç proje uygula
0 0 0 1 bir proje uygula	0 1 0 1	1 1 0 1
0 0 1 0	1 0 0 1	1 0 1 1
0 1 0 0	0 1 1 0	0 1 1 1
1 0 0 0	1 0 1 0	1 1 1 1 dört proje uygula
	1 1 0 0	

En iyi çözümü kesin olarak bulabilmek için, örneğimizde, 16 olası çözümü hesaplamamız gerekmektedir. Bu örnek TP için genel bir doğruyu gösterir. Küçük bir sorunu kolay bir şekilde çözmek için kullanılacak algoritma, sorun büyüdükçe artan bir hızla zorlaşmaktadır.

Örneğin ikili değerler (0-1) alabilen 100 değişkenli bir TP modeli için tek tek sayılacak olası çözüm sayısı 2x2x2x ... x2 = 2^{100} (yaklaşık olarak 10^{30}) adettir Tüm olası çözümleri saymak ve olurlular içinden en iyisini seçmek ***kavramsal*** olarak bir sorun yaratmasa da ***işlemsel*** (sayısal) olarak imkansızdır.

Dal Sınır

Sistematik bir şekilde olurlu çözümlerin sayılarak en iyi tamsayılı çözümün bulunması için kullanılan genel amaçlı bir DP tabanlı ağaç arama (LP-based tree search) yöntemi olan ***Dal Sınır Algoritması*** 1960'ların başında Land ve Doig tarafından önerilmiştir.

Sayma yöntemi gibi tüm olurlu çözümleri saymak yerine sadece belirli sayıda olurlu çözümü inceleterek (küçük bir kısmının inceleneceği ümidi ile) en iyi çözümü garanti bir şekilde bulur.

Örneğin çok dönemli sermaye bütçeleme örneğini ele alalım:

Enb. $0.2\,x_1 + 0.3\,x_2 + 0.5\,x_3 + 0.1\,x_4$

Öyle ki $0.5\,x_1 + 1\,x_2 + 1.5\,x_3 + 0.1\,x_4 \le 3.1$

$0.3\,x_1 + 0.8\,x_2 + 1.5\,x_3 + 0.4\,x_4 \le 2.5$

$0.2\,x_1 + 0.2\,x_2 + 0.3\,x_3 + 0.1\,x_4 \le 0.4$

$x_j = 0$ veya 1 $j = 1, \ldots 4$

Bu sorunun çözümünü güçleştiren değişkenlerin tamsayı değerler (sıfır veya bir) almaya zorlanmasıydı.

Eğer değişkenler kesirli değerler alabilseydi (sıfır ve bir arasındaki tüm değerler) sorunu DP olarak çözebilirdik.

Sorunun bu şekilde bir DP gevşetmesini [x_j = 0 veya 1 (j=1,...,4) yerine 0 <= x_j <= 1 (j=1,...,4) kısıtlarını kullanalım] çözeceğimizi varsayalım.

Herhangi bir DP çözüm paketi ile x_2=0.5, x_3=1, x_1=x_4=0 ve en iyi amaç fonksiyon değeri 0.65 olarak hesaplanır.

Bu durumda esas tamsayılı sorunumuzun en iyi amaç fonksiyon değeri hakkında fikir sahibi oluruz (<= 0.65). 0.65 en iyi tamsayılı çözüm için bir ***üst sınır***dır (upper bound).

Söz konusu değerin üst sınır olmasının nedeni tamsayı zorunluluğunu kaldırınca elde edilen DP gevşetmesinin çözümünün değerinin en az tamsayılı çözümünki kadar (belki de daha iyi) olacağıdır.

Elde edilen DP gevşetmesi çözümünde x_2 değişkeninin tamsayılı yerine kesirli bir değer aldığını görürüz. Tamsayılı değer alması için iki yeni sorunun çözümü ile ilgilenebiliriz:

P1: orijinal DP gevşetmesine ek olarak $x_2=0$

P2: orijinal DP gevşetmesine ek olarak $x_2=1$

Bu şekilde sadece bir değişkenin kesirli değer yerine tamsayılı değer almaya zorlanarak yeni çözüm aranması işlemine ***dal işlemi*** (branching) denilir. Dal işlemi, bir ağaç üzerinde (ağaç arama isminin nasıl ortaya çıktığını gösterecek şekilde) şematize edilebilir.

P1 ve P2 DP gevşetmelerini çözersek, aşağıdaki çözümleri elde ederiz:

P1'in çözümü $x_1=0.5$, $x_3=1$, $x_2=x_4=0$, $z= 0.6$

P2'in çözümü $x_2=1$, $x_3=0.67$, $x_1=x_4=0$, $z=0.63$

Şekilden de görüldüğü gibi yeni çözümlerdeki değişken değerleri de tamsayılı değildir:

En iyi tamsayılı çözümü bulmak için süreci tekrar ederiz, kesirli değer alan bir değişkeni tamsayılı olmaya zorlayarak iki yeni sorun üretiriz..

P1 sorunu üzerinde dal işlemi yapalım. Bu durumda elde edilecek DP gevşetmeleri ve çözümlerinin listesi:

P3 (P1'e ek olarak $x_1=0$) çözümü $x_3=x_4=1$, $x_1=x_2=0$, amaç fn.: 0.6

P4 (P1'e ek olarak $x_1=1$) çözümü $x_1=1$, $x_3=0.67$, $x_2=x_4=0$, amaç fn. 0.53

Görüldüğü gibi bu adımda P3 sorununun çözümü 0.6'dır ve tüm değişkenlerin değerleri tamsayılıdır. P3 üzerinde yeni bir dal işlemi yapmaya gerek kalmaz ve DP gevşetmeleri listesinden çıkarılır.

En iyi tamsayılı çözüm değeri hakkında yeni bir bilgi elde etmiş oluruz: amaç fonksiyonunun en iyi çözüm değeri 0.6 ile 0.65 arasında (her iki değer dahil) olacaktır.

P4'ü düşünelim, şu anki değeri 0.53'dür ve x_3 değişkeni kesirlidir. Yeni bir dal işlemi yapmamız durumunda amaç fonksiyon değeri daha iyi duruma gelemez. Zaten 0.6'lık bir tamsayılı çözümümüz olduğundan P4 de DP gevşetmeleri listesinden çıkarılır. Bu şekilde ***alt***

sınırdan (lower bound) daha iyi bir olurlu çözüm bulunamayacağı için eleme yapılmasına ***sınır işlemi*** (branching) denilir.

Geriye sadece P2 kalır:

P2 çözümü x_2=1, x_3=0.67, x_1=x_4=0, amaç fn: 0.63

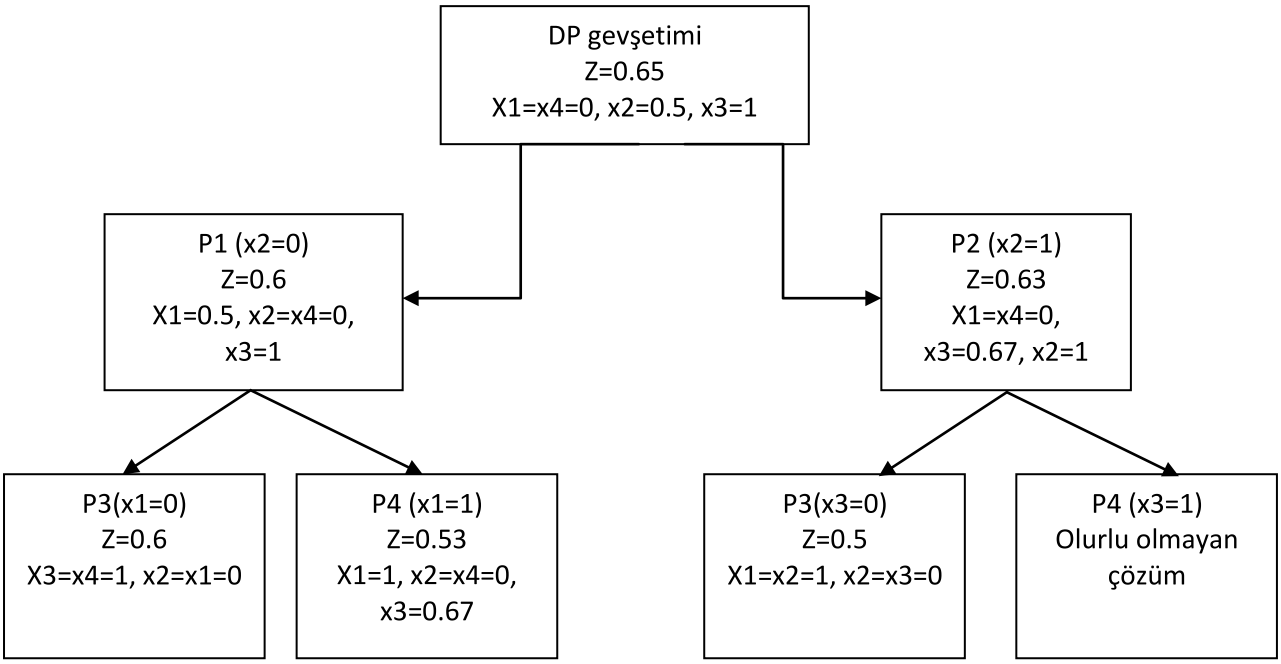
P2 için dal işlemi yapılırsa

P5 (P2'ye ek olarak x_3=0) çözümü x_1=x_2=1, x_3=x_4=0, amaç fn: 0.5

P6 (P2'ye ek olarak x_3=1) çözümü olurlu değil

Bu durumda ne P5 ne de P6'dan dal işlemi yapmak gerekmez.

En iyi tamsayılı çözüm (esas sorunun çözümü) amaç fonksiyon değeri 0.6 ve x_3=x_4=1, x_1=x_2=0 şeklindedir.

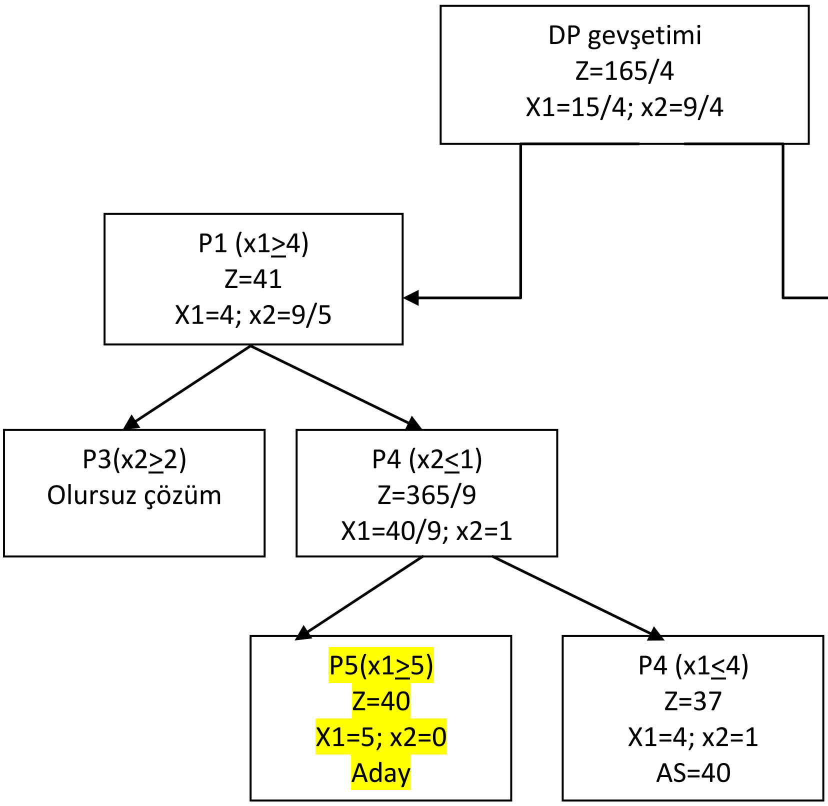


En iyi çözümü elde etmek için kullanılan tüm süreç:

Dikkat edileceği gibi sayma yönteminde sayılacak tüm olası çözümler 16 (2^4) adet iken biz sadece 7 DP çözdük. Dal sınır algoritması çok büyük sorunlarla ilgilenilmemesi durumunda en iyi tamsayılı çözümü bulmak için etkin bir yoldur.

Örnek:

Enb z=8x1+5x2
Öyle ki:
x1+x2<6
9x1+5x2<45
x1,x2>0 ve tamsayı

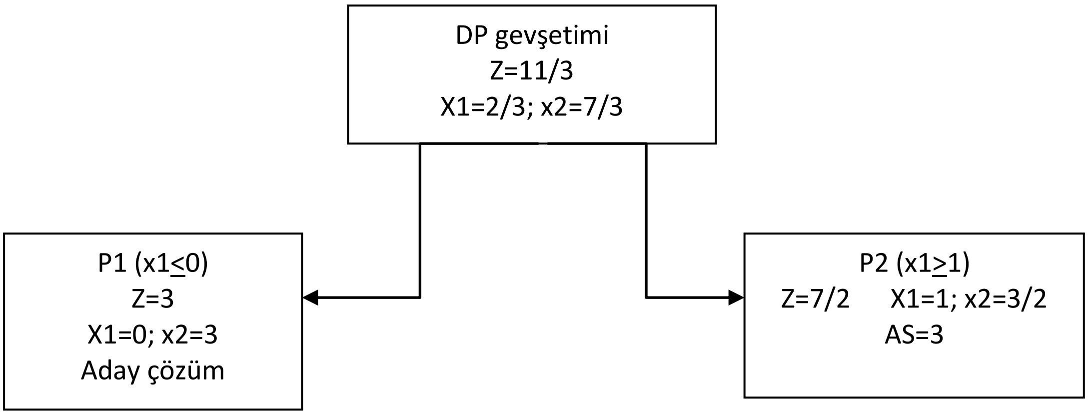


Karışık TS problemlerinin çözümü

Bu tarz problemlerin dal sınır algoritması ile çözümünde sadece tamsayı değişkenler üzerinden dallanma yapılır.

Örnek:

Enb z = 2 x1+ x2
Öyle ki
5 x1 + 2 x2 <= 8
x1+ x2<= 3
x1, x2 . 0; x1 tamsayı



2.alt problem x1'in tamsayı olma gerekliliğini yerine getirmekte ve 1.alt problemden daha iyi bir çözüm vermektedir. Dolayısıyla en iyi çözüme ulaşılmıştır.

Sırt çantası problemlerinin çözümü

Tüm değişkenlerin 0-1 tamsayı olduğu tek kısıtlı problemlerin geneline sırt çantası problemleri dendiğini biliyoruz.

Problemin yapısı gereği, DP gevşetimi için grafik çözüm ya da herhangi bir yazılıma gerek kalmaz. Şöyle ki:

ci i.değişkenden sağlanan yarar

B eldeki kaynak miktarı

ai ise i.değişkenin eldeki kaynağı kullanım miktarı olmak üzere, bir sırt çantası problemi:

Enb $z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$
Öyle ki $a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b$

$x_i = 0$ or 1 ($i = 1, 2, \ldots, n$) olarak ifade edilebilir.

(ci/ai) oranı (ri olarak adlandıralım) i.değişkenin kullandığı her kaynak miktarın başına sağlanan yarar olarak ifade edilebilir. Böylece, daha büyük r'ye sahip değişkenlerin en faydalı/iyi oldukları söylenebilir.

Bir sırt çantası problemini çözmek için, tüm değişkenler için bu oranı hesaplamak gerekir. Daha sonra en iyi orana sahip değişken sırt çantasına koyulmalıdır. Ve oran sırasına bağlı olarak sırt çantasına yerleştirme devam etmelidir.

Örnek

Enb $z = 8 x_1 + 11 x_2 + 6 x_3 + 4 x_4$
Öyle ki
$5 x_1 + 7 x_2 + 4 x_3 + 3 x_4 \leq 14$
$x_j = 0$ or 1 $j = 1, \ldots 4$

Oranları hesaplayalım:

$r_1 = 8 / 5 = 1.6$
$r_2 = 11 / 7 = 1.57$
$r_3 = 6 / 4 = 1.5$
$r_4 = 4 / 3 = 1.33$

Bu oranları kullanarak problemin DP gevşetimi uygulanmış çözümü:

$x_1 = 1$, $x_2 = 1$, $x_3 = 0.5$, $x_4 = 0$, $z = 22$

x_3 üzerinden dallanırız

P1 (DP gevşetimi+ $x_3=0$): $x_3=0$, $x_1=x_2=1$, $x_4=2/3$, $z=21.67$

P2 (DP gevşetimi + $x_3=1$): $x_3=x_1=1$, $x_2=5/7$, $x_4=0$, $z=21.85$

P2'yi seçip buradan ilerleyelim. Bu durumda x_2 üzerinden dallanırız :

P3 (P2 + $x_2=0$): $x_3=1$, $x_2=0$, $x_1=1$, $x_4=1$, $z=18$

P4 (P2 + $x_2=1$): $x_3=x_2=1$, $x_1=3/5$, $x_4=0$, $z=21.8$

P3 olurlu bir çözümdür. Yani aday çözümdür. Öyleyse: AS = 18 olur

P4 üzerinden ilerlenir ve x_1 üzerinden dallanma yapılırsa :

P5 (P4 + $x_1=0$): $x_3=x_2=1$, $x_1=0$, $x_4=1$, $z=21$ olur. Aday çözümdür. AS=21 olur

P6 (P4 + x1=1): Olursuz çözümdür

Bu aşamada P1 alt problemine dönersek, z değerinin 21.67 olduğu görülür ki buradan ilerlendiğinden hali hazırda bulunan z değerinden daha iyi bir sonuç vermesi mümkün değildir. O yüzden bu aşamada o dal budanır.

Optimal çözüm: P5'de bulunan çözümdür:

x3=x2=1, x1=0, x4=1, z=21