Dokuz Eylül Üniversitesi
Sosyal Bilimler Enstitüsü Dergisi
Yayın Geliş Tarihi: 01.10.2010 Cilt: 12, Sayı: 4, Yıl: 2010, Sayfa: 61-74
Yayına Kabul Tarihi: 17.01.2011 ISSN: 1302-3284

ÇOK AMAÇLI DOĞRUSAL PROGRAMLAMADAN SİSTEM TASARIMINA: DE NOVO

Kaan YARALIOĞLU [*]
Nurullah UMARUSMAN[**]

Özet

Bu çalışmada Çok Amaçlı Doğrusal Programlamadan De Novo Programlamaya geçiş bir süreç içinde tanımlanmaya çalışılmıştır. Bu şekilde sistem tasarımının önemi, verilen bir sistemin optimizasyonunun yerine bir optimal sistem tasarımının belirlenmesi farklılığı açıklanarak vurgulanmıştır. Çalışmanın bütünselliği ise tek bir örnek problem kullanılarak sağlanmıştır.

Anahtar Kelimeler: *Çok Amaçlı Doğrusal Programlama, Uzlaşık Programlama, De Novo Programlama*

FROM MULTIPLE OBJECTIVE LINEAR PROGRAMMING TO SYSTEM DESIGN: DE NOVO

Abstract

In this study, it is attempted to delineate the stages of the switching process from Multiple Objective Linear programming to De Novo programming. In this fashion, the import of system design is highlighted by expounding the distinctiveness of identifying an optimal system design instead of optimizing a given system. The integrity of the study has been ensured by means of a single sample.

Key Words: *Multiple Objective Linear Programming, Compromise Programming, De Novo Programming*

1. AMAÇ

İlk geliştirildiğinden bu yana klasik Doğrusal Programlama, belirli bir amacın belirli kısıtlar altında optimum koşullar altında gerçekleşmesini sağlayan değişken değerlerini veren bir yöntem olarak tanımlanmış ve kullanılagelmiştir. Ancak hayatın tek bir amaçla sınırlandırılamayacak kadar karmaşık olması,

[*] Doç. Dr., Dokuz Eylül Üniversitesi İİBF, Ekonometri Bölümü, +90 232 412 07 23, k.yaralioglu@deu.edu.tr.
[**] Yrd. Doç. Dr., Aksaray Üniversitesi, İİBF, İktisat Bölümü, +90 382 288 24 51 nurullah.umarusman@aksaray.edu.tr.

yöntemin sadece tek bir amacı en uygun koşullarda sağlamaya yönelik yapısını zorlamış ve zaman içinde klasik Doğrusal Programlama temel algoritması korunarak geliştirilmiştir.

Birden fazla amacın varlığında Doğrusal Programlama her amaç için ayrı ayrı çözülecektir. Ancak doğaldır ki her amaç için farklı bir çözüm seti oluşacak bu da uygun olmayan çözüm alanını ortaya çıkaracaktır.

Özellikle Hedef Programlama birçok amacı aynı anda değerlendirilebilen ve amaçlardan sapmaları minimize edebilen yapısıyla, bu alanda ve anlamda fayda yaratmıştır. Aynı zamanda Hedef Programlamanın uygun olmaya çözümlere yönelik Uzlaşık Programlama versiyonun da önemli bir boşluğu doldurduğu söylenebilir.

Ancak amaç ya da amaçlar her ne şekilde düzenlenirse düzenlensin, temel sorun değişken değerlerinin belirlenmesidir. Bu aşamada kaynaklar sadece amacın gerçekleşmesini sınırlayıcı etmenler olarak kabul edilirler. Yani başlangıçta verilen kısıtlar altında, kısıt değerlerinin sabitlenmiş bir değer seti çerçevesinde belirli bir amaç başarılmak istenir. Süreç sonunda ise çoğunlukla kullanılmayan veya fazla kaynak kapasitesi**,** hammadde ihtiyacı ortaya çıkar. Ama amaç bir şekilde gerçekleştirilmiştir.

Beklenen bu üretim belirsizliği karar vericiler için hem kaynak açısından üretimin optimum olmasına engel teşkil eder hem de beklenen kar açısından azalmalara yol açar. Oysa modern karar vericiler açısından artık önemli olan, maksimum kar ya da minimum maliyet düşüncesi değildir. Onlar eldeki kaynakları tam kapasiteyle kullanarak optimum üretim modelinin oluşturulmasının sağlamaya çalışırlar (Babic ve Pavic, 1996).

Bu anlamda De Novo Programlama da karar vericiler için bir çözüm alanı olarak ortaya çıkmaktadır. De Novo Programlama, verilen bir sistemin optimal tasarımının yerine bir optimal sistemin belirlenmesi üzerine yoğunlaşmaktadır (Bare ve Mendoza, 1989).

Bu çalışmanın yukarıda anlatılanlar çerçevesindeki temel amacı, Çok Amaçlı Doğrusal Programlamadan sistem tasarımına kadar olan bir anlamda tarihsel süreci aşamalandırmak ve her aşamanın bir öncekiyle olan ilgisini tanımlamak ve sonuçta De Novo Programlamanın karar vericilere sağladığı kazanımları ortaya koymak olarak belirlenmiştir.

2. YÖNTEM

Bu çalışma yukarıda belirtilen amacı dört aşamalı bir süreçte ve tek bir örnek kullanarak gerçeklemeye çalışmıştır. Aşamalar, Çok Amaçlı Doğrusal Programlama, Uzlaşık Programlama ve De Novo Programlama olarak belirlenmiştir.

Her bir aşamadaki yöntem öncelikle modellenmiş, ana hatlarıyla açıklanmış ve söz konusu örnek, yöntem üzerinde çözülmüştür. Çözüm bir önceki yöntemin çözümüyle karşılaştırılmış ve bir önceki yöntem üzerine getirdiği faydalar ortaya konmuştur. Ayrıca her aşamada bir sonraki yöntemi gerekçelendirmek üzere, karar vericiler için sorun oluşturacak noktalar belirlenmiştir.

Bu şekilde son yöntem çözümlendiğinde, karar vericilerin için tüm yöntemleri karşılaştırabilecekleri bir özet tablo ortaya konmuştur. Bu özet tablo aynı zamanda Doğrusal Programlamadan Sistem Tasarımına bir yol haritası oluşturması amacı taşıyarak hazırlanmaya çalışılmıştır.

Sözü edilen örnek aşağıda verilmiştir. Örnek maksimizasyon yönlü üç amaç (kar, kalite, müşteri memnuniyeti), üç değişken (üretilmesi düşünülen üç ürün) ve altı kısıt (freze, torna, bileyici, testere, matkap ve şerit testere makinelerinde kullanılabilecek zaman) içermektedir. Ayrıca her aşamada, bir sonraki aşamanın ortaya çıkmasını gerektirecek veri seti ilaveleri ayrıca yapılmıştır.

$$Z_1 = 50x_1 + 100x_2 + 17.5x_3$$

$$\text{Maks } Z_2 = 92x_1 + 75x_2 + 50x_3$$

$$Z_3 = 25x_1 + 100x_2 + 75x_3$$

Kısıtlar,

$$12x_1 + 17x_2 \leq 1400$$

$$3x_1 + 9x_2 + 8x_3 \leq 1000$$

$$10x_1 + 13x_2 + 15x_3 \leq 1750$$

$$6x_1 + 16x_3 \leq 1325$$

$$12x_2 + 7x_3 \leq 900$$

$$9.5x_1 + 9.5x_2 + 9.5x_3 \leq 1075$$

$$x_i \geq 0, i = 1,2,3$$

3. BÜTÇE KAVRAMI

Klasik Doğrusal Programlama ya da Doğrusal Programlama bazlı diğer programlama tekniklerinde temel hedef, amaç ya da amaçları gerçekleştirecek değişken değerlerinin elde edilmesidir. Bu yolda kısıtları oluşturan kaynakların, eğer amaç doğrudan bu değilse, ne kadar verimli kullanıldığı ya da kullanılmadığı genelde ikinci planda kalmaktadır.

Sonuçta, değişken değerleri amacı en iyiler ve bu değerlerin ötesinde amacı daha iyiye götürecek başka bir değer seti mevcut değildir. Ancak bu durum, tüm kaynakların verimli kullanıldığı anlamına gelmez. Diğer bir deyişle amacı en iyileyen çözüm, kaynakların önemli bir kısmının atıl kalması ya da tersine önemli miktarda eksik kaynak durumunu gerektirebilir.

Kaynakların verimli kullanılıp kullanılmadığı ise bütçe kavramıyla açıklanabilir. Burada başlangıç bütçesi, gereken kaynak miktarıyla kaynaklara ilişkin birim fiyatların çarpımından oluşacaktır. Çözüm sonucunda kullanılan kaynak miktarıyla birim fiyatların çarpımı, gerçekleşen bütçeyi oluşturur. Başlangıç bütçesiyle gerçekleşen bütçenin birbirine eşit olması ise kaynakların tam verimli olarak kullanıldığını gösterecektir (Bare ve Mendoza, 1990).

Bu noktada gerçek anlamda bir optimizasyonun, sadece amaçları gerçekleyen değil aynı zamanda kaynakları yani bütçeyi etkin olarak kullanan bir program olduğu söylenebilir.

Yukarıdaki örneğe ilişkin 6 kısıt için birim fiyatlar ve bütçe Tablo 1'de gösterilmiştir. Bu örnek için bütçenin, 4658,75 $ olduğu görülebilir.

Tablo 1: Örneğe İlişkin Bütçe

Kaynaklar (b_i)	Kaynakların Birim Fiyatı (p_i)	Toplam
b_1 = 1400	p_1 = 0,75 $	1050 $
b_2 = 1000	p_2 = 0,60 $	600 $
b_3 = 1750	p_3 = 0,35 $	612,5 $
b_4 = 1325	p_4 = 0,50 $	662.5 $
b_5 = 900	p_5 = 1,15 $	1035 $
b_6 = 1075	p_6 = 0,65 $	698,75 $
$\text{Bütçe}: \sum_{i=1}^{6} p_i . b_i$		4658,75 $

4. ÇOK AMAÇLI DOĞRUSAL PROGRAMLAMA

Çok Amaçlı Doğrusal Programlama, iki veya daha fazla amaç fonksiyonunu içeren optimizasyon tekniğidir ve klasik (tek amaçlı) Doğrusal Programlamadan farkı, sadece amaç fonksiyonlarının yapısından meydana gelmektedir. Tek amaçlı Doğrusal Programlamada çözümün hedefi, amaç fonksiyonunun en iyi değerini veren değişkenlerin belirlenmesidir. Bu sebeple amaç fonksiyonunun optimum değeri tektir. Diğer yandan, Çok Amaçlı Doğrusal programlama birden fazla maksimizasyon ve minimizasyon yönlü amaçların her ikisine de sahip olabilir.

Çok amaçlı doğrusal programa tümüyle doğrusal programlamanın varsayımlarına sahiptir ve çözümü yine simplex yöntemle gerçekleştirilir. Ancak çözüm aynı kısıt setinde her bir amaç için ayrı ayrı aranır.

Bu da doğal olarak, her bir amacı gerçekleştirecek şekilde farklı değişken değerlerinin (çözüm değerleri) ortaya çıkması anlamına gelecektir. Bu durum ise karar verici açısından yeni bir karar sorununu ortaya çıkaracaktır.

Çok Amaçlı Doğrusal Programlamada kısıtların sınırlı miktarları ile amaç fonksiyonları arasında farklı birim ve yapıların olması beklenen bir durumdur. Yani bütün amaç fonksiyonlarının eşzamanlı olarak optimum durumu hemen hemen imkânsızdır (Romero, 1985).

Bu çalışmada tanımlanan örneğin çözüm sonuçları Tablo 2'de verilmiştir.

Tablo 2: Çok Amaçlı Doğrusal Programlama Çözümü

	1. Amaç	2. Amaç	3. Amaç
Amaç Değeri	$Z_1 = 8041.14$	$Z_2 = 10950.59$	$Z_3 = 9355,89$
Çözüm	$x_1 = 44,93$ $x_2 = 50,63$ $x_3 = 41,77$	$x_1 = 92,97$ $x_2 = 0$ $x_3 = 47.95$	$x_1 = 45,22$ $x_2 = 49,61$ $x_3 = 43,53$

Çalışmada yer verilen örnek beklendiği gibi her amaç için farklı sonuç seti vermiştir.

5. UZLAŞIK PROGRAMLAMA

Örneğin çözümünden Tablo 2'den de görüleceği ve beklendiği gibi tek bir çözüm seti elde edilmemiştir. Bu uygun olmayan çözüm durumunu göstermektedir. Eğer çok sayıda amaç için bir optimal çözüm aranıyorsa, Uzlaşık Programlama yaklaşımı kullanılmaktadır.

Her bir amaç için farklı çözümler (uygun olmayan çözümler), üstün olmayan çözüm durumu olarak ifade edilmektedir. Buradaki sorun üstün olmayan çözümler ile ideal çözüme yakınlığın ortaya konabilmesidir. Eğer üstün olmayan çözümler ile ideal çözümler arasındaki yakınlık minimize edilebilirse iki çözüm arasında bir uzlaşı sağlanabilecektir. Bu süreci gerçekleştiren yaklaşım ise uzlaşık programlama olarak adlandırılmaktadır (Huang, Tzeng ve Ong, 2005).

Uzlaşık programlama 1973 yılında P. Yu ve M. Zeleny tarafından geliştirilmiş doğrusal programlama tabanlı bir yöntemdir. Yöntemin temel amacı, çok sayıdaki amaç fonksiyonları arasındaki d ile gösterilen sapmayı minimum kılacak bir çözüm setini elde etmektir. Burada sapma değeri $0 \le d \le 1$ aralığındadır ve sapma değerinin 0'a yakın olması ideal çözüme yakınlığı gösterir.

Uzlaşık programlama yönteminin adımları aşağıda gösterilmiştir (Zeleny, 1982);

Adım 1: Amaç fonksiyonu oluşturulur

Uzlaşık programlama amaçlardan sapmayı minimize eder. Bu nedenle amaç fonksiyonu, formül 1'de gösterildiği gibi sadece sapma değişkeninden meydana gelir.

$$\min Z = d \tag{1}$$

Adım 2: Pozitif ve negatif ideal çözüm kümeleri oluşturulur.

Uzlaşık programlamada öncelikle her bir amaç için ve amacın yönüne bakılmaksızın, amaç fonksiyonunun çözüm değerleri bulunur. Pozitif ideal çözüm kümesi (I^*) formül 2'de gösterildiği üzere amaç fonksiyonun karar sorunundaki amaç yönlerine ilişkin çözüm değerleridir. Burada Y_k^*, k adet maksimizasyon yönlü amaç çözüm değerini, W_l^* ise l adet minimizasyon yönlü amaç çözüm değerini gösterir.

$$I^* = \{Y_1^*, Y_2^*, ..., Y_k^*; W_1^*, W_2^*, ..., W_l^*\} \tag{2}$$

Negatif ideal çözüm kümesi (I^-) ise amaç fonksiyonlarının başlangıçtakinden ters yönlü çözüm değerlerinden oluşur. Negatif ideal çözüm kümesi formül 3'de gösterilmiştir.

$$I^- = \{Y_1^-, Y_2^-, ..., Y_k^-; W_1^-, W_2^-, ..., W_l^-\} \tag{3}$$

Adım 3: Sapma kısıtları oluşturulur.

Bir uzlaşık programlama modelinde amaç sayısı kadar sapma kısıtı vardır. Sapma kısıtları, maksimizasyon yönlü amaçlar için formül 4'den, minimizasyon yönlü amaçlar için formül 5'den yararlanarak elde edilir.

$$d \geq \frac{Y_k^* - \max Z_k}{Y_k^* - Y_k^-} \tag{4}$$

$$d \geq \frac{\min Z_l - W_l^-}{W_l^- - W_l^*} \tag{5}$$

Uzlaşık Programlama yaklaşımı için literatürde dört farklı yöntem önerilmektedir. Bu yöntemler aşağıda sıralanmıştır;

- Utility (Fayda) Yaklaşımı
- Hedef Programlama
- İnteraktif Yaklaşım

- Bulanık Yaklaşım

Bu çalışmada uzlaşık çözüm için minmaks hedef programlama yaklaşımı kullanılmıştır. Minmaks hedef programlamaya göre yeni model aşağıdaki gibi yazılabilir;

$$Min \ \ d$$
$$50x_1 + 100x_2 + 17.5x_3 + n_1 = 8041.14$$
$$92x_1 + 75x_2 + 50x_3 + n_2 = 10950.59$$
$$25x_1 + 100x_2 + 75x_3 + n_3 = 9355.895$$
$$\frac{n_1}{d} \le 8041.14$$
$$\frac{n_2}{d} \le 10950.59$$
$$\frac{n_3}{d} \le 9355.59$$
$$12x_1 + 17x_2 \le 1400$$
$$3x_1 + 9x_2 + 8x_3 \le 1000$$
$$10x_1 13x_2 + 15x_3 \le 1750$$
$$6x_1 + 16x_3 \le 1325$$
$$12x_1 + 7x_3 \le 900$$
$$9.5x_1 + 9.5x_2 + 9.5x_3 \le 1075$$
$$x_i \ge 0, i = 1,2,3$$

Modelde, amaçtan negatif sapma (d) minimize edilecektir. Ayrıca burada Çok Amaçlı Doğrusal Programlama modelindeki amaçlar kısıt haline getirilmiştir. Söz konusu kısıtlarda bir önceki aşamada (Çok Amaçlı Doğrusal Programlama) bulunan amaç değerlerini gerçekleştirilmeye çalışılacaktır. Modelin çözümünde ise bu amaçları minimum toplam negatif sapmayla gerçekleştirecek değişken değerleri elde edilecektir. Burada n_i değişkenleri ilgili amaç için negatif sapmayı, d değeri ise örnek için üç amaçtaki toplam minimum negatif sapmayı göstermektedir.

Modelin çözümü ve çok amaçlı doğrusal programlama çözümü ile karşılaştırması Tablo 3'te verilmiştir.

Tablo 3: Çok Amaçlı Doğrusal Programlama ve Uzlaşık Çözüm Karşılaştırması

Çok Amaçlı Doğrusal Programlama Çözümü	Uzlaşık Çözüm
Uygun Olmayan Çözüm	$b_1 = 1327{,}62$ $b_2 = 892{,}68$ $b_3 = 1749{,}85$ $b_4 = 1056{,}36$ $b_5 = 808{,}35$ $b_6 = 1074{,}92$
$\sum_{i=1}^{6} p_i . b_i = 4658{,}75\$$	$\sum_{i=1}^{6} p_i . b_i = 4300\ \$$
$Z_1 = 8041{,}14$ $x_1 = 44{,}93$, $x_2 = 50{,}63$, $x_3 = 41{,}77$	$Z_1 = 7518{,}87$ $x_1 = 53{,}26$, $x_2 = 40{,}5$, $x_3 = 46{,}05$
$Z_2 = 10950{,}59$ $x_1 = 92{,}97$, $x_2 = 0$, $x_3 = 47.95$	$Z_2 = 10239{,}92$ $x_1 = 53{,}26$, $x_2 = 40{,}5$, $x_3 = 46{,}05$
$Z_3 = 9355{,}89$ $x_1 = 45{,}22$, $x_2 = 49{,}61$, $x_3 = 43{,}53$	$Z_3 = 8835{,}25$ $x_1 = 53{,}26$, $x_2 = 40{,}5$, $x_3 = 46{,}05$

Tablo 3'ten görüleceği gibi uzlaşık çözüm ile uygun olmayan çözüm durumu giderilmiş üç amacı da gerçekleyecek üretim değerlerine ulaşılmıştır. Sapma değeri d = 0,06483775 elde edilmiştir. Ancak burada, diğer Doğrusal Programlama versiyonlarında da görüldüğü gibi, kaynakların etkin olarak kullanılamadığı (sadece 3. kaynak (bileyici) ve 6. kaynak (şerit testere) zamanları tam olarak kullanılmıştır) görülmektedir.

Her ne kadar çözümün gerçekleşmesinin yanı sıra bütçede 358,75 (4658,75 – 4300) $'lık bir tasarruf görülse de amaç değerlerindeki düşmeler, akılcı karar verici için kabul edilebilir değildir.

Bu noktada akılcı karar verici için gerçek optimal çözümün, hem kaynakların daha etkin kullanıldığı hem de daha yüksek amaç değerlerine sahip olduğu bir çözüm seti olduğu söylenebilir.

6. DE NOVO PROGRAMLAMA

Klasik Doğrusal Programlama problemlerinde başlangıçta verilen kısıtlar altında, kısıt değerlerinin sabitlenmiş bir değeri çerçevesinde belirli bir amaç başarılmak istenir. Yani kısıtların ya da kaynakların verimli kullanımı amacın yanında ikinci derece öneme sahiptir. Diğer bir deyişle kısıtlar amacın belirleyicisidirler. Kısıt (kaynak) değeri açısından incelenen problemin çözümü sonucunda, çoğunlukla kullanılmayan veya fazla kaynak kapasitesi, hammadde ihtiyacı ortaya çıkar. Beklenen bu belirsizlik, karar vericiler için hem kaynak açısından üretimin optimum olmasına engel teşkil eder hem de beklenen kar açısından azalmalara yol açar. Oysa özellikle bu günün rekabetçi koşullarında karar vericiler açısından önemli olan, maksimum kar yada minimum maliyet düşüncesinden sıyrılıp eldeki kaynakları tam kapasiteyle kullanarak optimum karar modelinin oluşturulmasının sağlanmasıdır (Yaralıoğlu, 2004).

Karar süreçlerinde planlama dönemlerinin kısalması yani kısa vadede karar verme zorunluluğu, sahip olunan kaynakları en az amaçlar kadar önemli hale getirmiştir. Bunun için de kısa vadede mevcut kaynakların bazıları sabit olsa bile, optimal şartlar için kaynaklar uzun vadede ya da sonraki planlama safhasında değiştirilmeli ve yeniden yapılandırılmalıdır. Kaynak miktarlarının optimum seviyede belirlenememesi, yeterli optimizasyonun sağlanamamasına ve kıt kaynakların verimli kullanılmamasına yol açar (Yaralıoğlu, 2010).

Bu anlamda De Novo Programlamanın geliştirilmesi, kaynakların uzun vadede yeniden yapılandırılmasına, kıt kaynakların daha verimli kullanılmasına ve sistemlerdeki savurganlığı önleyerek optimal tasarıma imkân sağlayan bir yöntem olarak karar vericiler için önemli bir boşluğu doldurmuştur.

De Novo Programlama, verilen bir sistemin optimal tasarımının yerine bir optimal sistemin belirlenmesi üzerine yoğunlaşmaktadır. Yani De Novo Programlama aynı zamanda bir sistem tasarımı tekniğidir (Li ve Lee, 1990).

De Novo Programlama kısıtların yeniden düzenlenmesine olanak sağlayarak sabitlenmiş kısıtlar altında ulaşılan çözümlerden daha uygun çözümler elde etmeye çalışır. Bu sebeple De Novo Programlama, verilen bir sistemi optimize etmek yerine amaçların başarılması mümkün olan en yüksek değerde ve kısıtların tam kapasite ile kullanılmasıyla bir optimal sistemin nasıl oluşturulması gerektiğini belirtir (Shi, 1999).

De Novo Programlamada kaynaklar, bütünleşik tek bütçe kısıtı kullanılarak sınırlandırılır. Bir başka deyişle, kaynakların maksimum miktarları bütçe tarafından yönetildiği için sınırlıdır. Bu ise De Novo Programlamanın en önemli öğesidir. Yani De Novo Programlamada kaynakların tam olarak verimli kullanılmasında belirleyici bütçe kavramı olmaktadır.

De Novo Programlama, analizin kaynaklar satın alınmadan önce yapıldığını varsayar. Çünkü analiz safhasında kaynaklar kontrol edilebilir ve henüz sabitlenmemiştir. Bunun yanında, kaynaklar bölünebilir olmalı ve istenilen miktarda satın alınmalıdır. Bu yaklaşımın özellikle önemli bir avantajı da, çok kriterli karar verme problemlerinde, kaynak kısıtlarının ayarlanmasına imkân vermesi ve başlangıçtaki ideal çözümü, aynı ya da daha düşük maliyetlerle daha uygun hale getirmeyi mümkün kılmasıdır.

De Novo çözümü bazı önemli özellikleri ortaya koyar. Yetersiz yararlanılan kaynak yoktur. Optimallik tanımından dolayı, tüm yapay değişkenler ve aylak değişkenler sıfıra eşit olmalıdır. Tüm kaynaklarını tam kapasite kullanmayan ve boşta kaynakları olan hiçbir sistem, optimal bir sistem olarak değerlendirilemez. Eğer yüksek verimli sistemlerden bahsetmek istiyorsak, tüm kaynaklar tam kapasite ile kullanılmalıdır.

Genel olarak De Novo Programlama modeli aşağıdaki gibi tanımlanmaktadır;

$$\text{Maksimize } Z = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j - b_i \leq 0$$

$$\sum_{i=1}^{m} p_i b_i \leq B$$

$$x_j \geq 0 ,\ j = 1,2,...,n ,\ i = 1,2,..,m$$

Klasik Doğrusal Programlama ve De Novo Programlama kısıt kaynaklarının yapısı ile ilgili olarak iki önemli sonuç vermektedir. Bunlar: Doğrusal Programlama problemlerinde kısıt kaynakların sabit ve kısıtlar katıdır. De Novo Programlamada ise bütün kısıt kaynaklarının esnekliği kaynakların yeniden tasarımlanabileceğine olanak tanımaktadır.

Sonuç olarak, sistem tasarımı, yeniden tasarımı ve optimizasyonu, sistem sınırlarının ve kısıtlarının amaca yönelik olarak yeniden şekillendirilmesini içermelidir. Sistem tasarımı, alternatiflerin bir seçimi değil, alternatiflerin yaratılması işlemidir.

Bu çalışmada kullanılan örnek De Novo Programlama kullanılarak modellendiğinde,

$$\text{Maks}\quad \begin{aligned} Z_1 &= 50x_1 + 100x_2 + 17.5x_3 \\ Z_2 &= 92x_1 + 75x_2 + 50x_3 \\ Z_3 &= 25x_1 + 100x_2 + 75x_3 \end{aligned}$$

Kısıtlar;

$$12x_1 + 17x_2 \leq b_1$$
$$3x_1 + 9x_2 + 8x_3 \leq b_2$$
$$10x_1 13x_2 + 15x_3 \leq b_3$$
$$6x_1 + 16x_3 \leq b_4$$
$$12x_1 + 7x_3 \leq b_5$$
$$9.5x_1 + 9.5x_2 + 9.5x_3 \leq b_6$$
$$0.75b_1 + 0.6b_2 + 0.35b_3 + 0.5b_4 + 1.15b_5 + 0.65b_6 \leq 4658.775$$

Modeldeki sonuncu kısıt bütçe kısıtıdır.

Ya da kısıtlar bütünleşik olarak yazıldığında model aşağıdaki gibi yazılabilir,

$$\text{Maks}\quad \begin{aligned} Z_1 &= 50x_1 + 100x_2 + 17.5x_3 \\ Z_2 &= 92x_1 + 75x_2 + 50x_3 \\ Z_3 &= 25x_1 + 100x_2 + 75x_3 \end{aligned}$$

$$23.475x_1 + 42.675.x_2 + 28.7x_3 \leq 4658.775$$

Modelin çözümü Tablo 4'te gösterilmiştir.

Tablo 4: De Novo Programlama Çözümü

	1. Amaç	2. Amaç	3. Amaç
Amaç Değeri	$Z_1 = 10916{,}81$	$Z_2 = 18257{,}93$	$Z_3 = 12174{,}43$
Çözüm	$x_1 = 0$ $x_2 = 109{,}17$ $x_3 = 0$	$x_1 = 198{,}46$ $x_2 = 0$ $x_3 = 0$	$x_1 = 0$ $x_2 = 0$ $x_3 = 162{,}33$

Tablo 4'ten görüleceği gibi çözüm sonucunda uygun olmayan çözüm gerçekleşmiştir. Bu Çok Amaçlı Doğrusal Programlama çözümünde olduğu gibi beklenen durumdur ve De Novo Programlama için de uzlaşık çözümün aranması gerekecektir.

Uzlaşık çözüm için model,

$$Min\ d$$
$$50x_1 + 100x_2 + 17.5x_3 + n_1 = 10916{,}81$$
$$92x_1 + 75x_2 + 50x_3 + n_2 = 18257{,}93$$
$$25x_1 + 100x_2 + 75x_3 + n_3 = 12174{,}43$$
$$\frac{n_1}{d} \leq 10916{,}81$$
$$\frac{n_2}{d} \leq 18257{,}93$$
$$\frac{n_3}{d} \leq 12174{,}43$$
$$23{,}475\ x_1 + 42{,}675\ x_2 + 28{,}7\ x_3 \leq 4658{,}775$$
$$x_i \geq 0,\ i = 1,2,3$$

Tablo 5: Çok Amaçlı Doğrusal Programlama, Uzlaşık Çözüm ve De Novo Karşılaştırması

Çok Amaçlı Doğrusal Programlama Çözümü	Uzlaşık Çözüm	De Novo Programlama Çözümü
Uygun Olmayan Çözüm	$b_1 = 1327{,}62$	$b_1 = 1465{,}06$
	$b_2 = 892{,}68$	$b_2 = 910{,}34$
	$b_3 = 1749{,}85$	$b_3 = 2030{,}5$
	$b_4 = 1056{,}36$	$b_4 = 1444{,}48$
	$b_5 = 808{,}35$	$b_5 = 640$
	$b_6 = 1074{,}92$	$b_6 = 1299.51$
$\sum_{i=1}^{6} p_i.b_i = 4658{,}75\$$	$\sum_{i=1}^{6} p_i.b_i = 4300\ \$$	$\sum_{i=1}^{6} p_i.b_i = 4658{,}59\ \$$
$Z_1 = 8041{,}14$ $x_1 = 44{,}93$ $x_2 = 50{,}63$ $x_3 = 41{,}77$	$Z_1 = 7518{,}87$ $x_1 = 53{,}26$ $x_2 = 40{,}5$ $x_3 = 46{,}05$	$Z_1 = 7688$ $x_1 = 92{,}48$ $x_2 = 20{,}9$ $x_3 = 55{,}6$
$Z_2 = 10950{,}59$ $x_1 = 92{,}97$ $x_2 = 0$ $x_3 = 47.95$	$Z_2 = 10239{,}92$ $x_1 = 53{,}26$ $x_2 = 40{,}5$ $x_3 = 46{,}05$	$Z_2 = 12857{,}5$ $x_1 = 92{,}48$ $x_2 = 20{,}9$ $x_3 = 55{,}6$
$Z_3 = 9355{,}89$ $x_1 = 45{,}22$ $x_2 = 49{,}61$ $x_3 = 43{,}53$	$Z_3 = 8835{,}25$ $x_1 = 53{,}26$ $x_2 = 40{,}5$ $x_3 = 46{,}05$	$Z_3 = 8572{,}5$ $x_1 = 92{,}48$ $x_2 = 20{,}9$ $x_3 = 55{,}6$

Modelin çözümü ve diğer iki çözüm ile karşılaştırması Tablo 5'te verilmiştir. Bu tablodan görüleceği gibi uzlaşık çözümde sadece iki kaynak (bileyici ve şerit testere) tam olarak kullanılırken De Novo çözümde tam kullanılan kaynak sayısı dörde (freze, bileyici, testere ve şerit testere) çıkmıştır. Tablo 5'ten çıkarılacak bir diğer sonuç ise Birinci ve ikinci amaçlar diğer iki durumdan daha yüksek değerlerde maksimize edilmiştir. Ayrıca bütçenin de tam olarak kullanıldığı yine Tablo 5'ten görülmektedir.

7. SONUÇ

Klasik Doğrusal Programlama çözümlerinde mevcut karar sistemi yalnızca planlanan ve başlangıçta verilmiş olan kısıtlar açısından değerlendirme yaparak amacın optimizasyonu ile ilgilenmektedir. Çözüm sonucunda çoğunlukla kaynakların niteliğine göre fazla kapasite miktarı veya daha fazla kaynak ihtiyacı ortaya çıkmaktadır. Bu durum, başlangıçta belirlenmiş olan kısıtların etkin olarak kullanılmamasından kaynaklanmaktadır.

Doğrusal Programlama modelinin yetersiz kaldığı bu durumda De Novo Programlama yaklaşımı yardımı ile hammadde kullanım kapasiteleri yeniden düzenlenebilir. Bu düzenleme ise mevcut hammadde için ayrılan yatırım bütçesine göre yapılmalıdır. Bu şekilde yapılan bir düzenleme ile kaynakların bütünüyle kullanımı sağlanmış olmakta ve amaçların daha yüksek bir değerde gerçekleştirilmesine imkân sağlanmaktadır.

Bu programlama çalışmalarının bütünü ise tam anlamıyla bir sistem tasarımı olarak değerlendirilebilir. Sonuçta karar sistemi, bir bütçe kavramı içinde ve doğrusal programlama mantığı kullanılarak yapılandırılmaktadır.

Bu çalışma kapsamında kullanılan örnek sonuçlarından da görüleceği gibi, De Novo Programlamanın avantajları aşağıdaki gibi özetlenebilir:

1. Kaynaklar daha etkin olarak kullanılabilmektedir,
2. Diğer programlama teknikleriyle kıyaslandığında aynı bütçeyle daha fazla kaynak kullanımına olanak sağlamaktadır,
3. Çok sayıda amaç için de uzlaşık çözüme olanak tanımaktadır,
4. Diğer programlama tekniklerine nazaran aynı kısıt değerleriyle amaçlar daha yüksek düzeylerde gerçekleştirilebilmektedir.
5. Diğer programlama teknikleriyle hemen hemen aynı varsayımları kullandığı için bu tekniklerin yerine tam ikame olanağı yaratmaktadır.

KAYNAKÇA

Babic, Z., Pavic, I. (1996). Multicriterial Production Planning By De Novo Programming Approach. *Introduction Journal of Production Economics*, 43 (1): 59-66.

Bare, B.B., Mendoza, G.A.(1989). Designing Forest Plans with Conflicting Objectives using De Novo Programming. *Journal of Environmental Management*, 31 (3): 237-246.

Bare, B.B. Mendoza, G.A. (1990). Log Allocation And Soft Optimization: A De Novo Programming Approach. *Forest Product Research Society*, 39 (9): 39-44.

Cohon, J. (1978). Multiobjective *Programming and Planning*, Academic Press, New York.

Huang J.J, Tzeng G.H., Ong, C.S. (2005). Motivation and Resource-Allocation For Strategic Alliance Through the De Novo Perspective. *Mathematical and Computer Modelling*, 41 (6-7): 711-721.

Li, R.J., Lee, E.S.(1990). Approaches To Multicriteria De Novo Programs. *Journal of Mathematical Analysis and Applications*, 153 (1): 97-111.

Romero, C.(1985). Multi-Objective And Goal Programming as A Distance Function Model. *Journal of Operational Research Society*, Vol. 36 (3): 249-251.

Schniederjans, M.J., Kwank, N.K.(1982). An Alternative Solution Method For Goal Programming Problems: A Tutorial. *Journal of Operational Research Society*, 33 (3): 247-251.

Shi, Y.(1999).Optimal System Design With Multiple Decision Makers and Possible Debt: A Multicriteria De Novo Programming Approach. *Operations Research*, 47 (5): 723-729.

Yaralıoğlu, K.(2004). *Uygulamada Karar Destek Yöntemleri*, İlkem Ofset, İzmir.

Yaralıoğlu, K.(2010) *Karar Verme Yöntemleri*, Detay Yayıncılık, Ankara.

Zeleny, M.(1982). Multiple Criteria Decision Making, McGraw-Hill, NewYork.