T.C. Dokuz Eylül Üniversitesi İktisadi ve İdari
Bilimler Fakültesi Dergisi, Cilt:4, Sayı:1-2, Yıl:1989

ETKİLEŞİMLİ BİR ÇOK AMAÇLI PROGRAMLAMA YÖNTEMİ

H.Hülya TÜTEK(x)

ÖZET

Bu çalışmada Nijkamp v.d. (1978) tarafından önerilen etkileşimli bir amaçlı programlama (EÇAP) yöntemi incelenmektedir. Yöntem, bir çözümleyiciyle karar vericinin bilgisayar aracılığıyla ard arda etkileşimde bulunmasına dayanmakta ve karar vericiden tercih ve önceliklerini aşama aşama sormaktadır. Basıc dilinde yazılan ve temel olarak doğrusal programlamaya ilişkin alt program kullanarak karar verici ile etkileşimi sağlayan bir program yardımıyla yöntem çok amaçlı hipotetik bir yatırım sorununa uygulanmaktadır.

GİRİŞ

Amaç programlama, çok amaçlı programlama modellerinin en güçlülerinin biridir. Uygulamadaki karar vermeye oldukça yaklaşmaktadır. Yöntem, firma davranış kuramı sonuçlarıyla uyum içindedir. Zira, uygulamada da karar vericiler belli hedeflere ulaşmayı gütmektedirler. Beklenti düzeyleri bu nedenle saptanır. Ancak, her hedefe ulaşma aynı derecede önemli değildir. Diğer bir deyişle , açıkça olmasa da yöneticiler değişik amaçlara değişik ağırlıklar verirler.

Amaç programlama Charnes ve Cooper (1961) tarafından amaç-kaynak etkileşiminden ortaya çıkan olursuz doğrusal programlama sorunlarına çözüm aranırken ortaya atılmış bir yöntemdir. Bu nedenle amaç programı doğrusal programlamanın bir uzanımıdır. Amaç programlama modelinin kurulması da doğrusal programlama modelinin kurulmasına benzer.(Spronk,1981).

(*) Doç.Dr. ,D.E.Ü .İ.İ.B.F. İşletme Bölümü

Amaç programlama modelinin kurulmasında ilk aşama değişkenlerin tanımlanmasıdır. İkinci aşamada tüm amaç değişkenler belirlenerek öncelik sırasına konur.Yönetim değişik amaçları kardinal ölçekte sıraya koyamazsa , ordinal bir sıralama yapabilir.Amaç programlama genellikle birbiriyle çelişkili çok amaçlı karar sorununa yönetim önceliklerine göre çözüm getirir.

Genel çoklu amaç programlama modeli matematiksel olarak aşağıdaki gibi tanımlanır:

en küçük $f(y_i^+ \ y_i^-)$

kısıtlar

$g_i(X) + y_i^+ - y_i^- = b_i$ (i=1,.........,m)

$X \in R$ olmak üzere

$R = \{ -X \mid h(x) < h_i \}$ (i=1,...........,m)

$y_i^+ . y_i^- = 0$ (i=1,..........,m)

$X, y_i^+, y_i^- >= 0$

Burada $f, g_i(X)$ amaç değişkenlerinin beklenti düzeyleri b_i değerlerinden y_i^+ pozitif ve y_i^- negatif sapmalarının bir fonksiyonudur.

İnstrumental X değişkenlerinin R olurlu alanı , h(X) kısıt seti ile sınırlandırılmıştır .Bu alan konvekstir.Genel olarak f'in de konveks olduğu varsayılmaktadır.Çoğu durumda hem amaç değişkenleri $g_i(X)$ hem de kısıtlar $h_i(X)$ dogrusaldır.Yani

$g_i(X) = A.X$

$h_i(X) = B.X$

Bu durumda doğrusal programlama modelinin temel varsayımları amaç programlama içinde geçerlidir.Ancak amaç programlamada doğrusal programlamada olduğu gibi bir amacın en büyüklemesi ya da en küçüklemesi söz konusu değildir.Yerine hedeflenen amaçlardan sapmaları amaçlara verilen önceliklere göre minimize eder.

Yukarıdaki sözü edilen uygulamaya uygunluğunun yanısıra yöntemin belli çekici özellikleri vardır. Bunlardan en önemlisi amaçların birbiriyle çelişkili durumlarda bile iyi tanımlanmış bir amaç programlama sorunu için, R olurlu alanının boş set olmaması koşuluyla, bir çözüm olmasıdır.Bu y_i^+ ve y_i^- sapma değişkenlerinin soruna katılmış olmasından kaynaklanmaktadır.Çoklu amaç programlamanın diğer bir önemli teknik avantajı çok karmaşık bir çözüm yordamı getirmemesidir.Özellikle doğrusal amaç programlama sorunları doğrusal programlama rutinleriyle çözülebilir.

Çoklu amaç programlamanın önemli bir sakıncası karar vericilerin tercihleri hakkında oldukça detaylı peşin (a priori) bilgi gerektirmesidir. Beklenti düzeylerinin belirlenmesi , öncelik sınıflarına bölünmesi ve bu sınıflara ağırlıklar atanması gerekmektedir. Bu nedenle çoğu yazarların etkileşimli amaç programlama kullanılmasını önermeleri doğaldır.Bu yöntemler , karar verici ile çözümleyicinin karşılıklı ve aşamalı olarak birbirleriyle alış verişine dayanmaktadır. Etkileşimli yöntemler , karar vericinin tercih fonksiyonunun açıkca belirlenmesini veya çelişen amaçları arasında takasların açıkca sayısal olarak saptanmasını gerektirmez . Etkileşimli yaklaşımda karar vericinin alternatif olurlu durumlar ile ilgili öncelikler konusunda bilgi vermesi gerekmektedir.

Etkileşimli yordamlarda karar vericiden yanlızca kısıtlı bilgi, yeri geldikçe aşama aşama sorulur. Çözümleyicinin görevi amaçların uygun değerleri, dolayısıyla uygun çözümlerle ilgili tüm bilgileri karar vericiye sunmaktır.Ne yazık ki , etkileşimli yöntemlerin çoğu , örneğin öncelikleri de içermek gibi "geleneksel " amaç programlama yöntemlerinin bazı avantajlarından yoksundurlar.

ETKİLEŞİMLİ ÇOKLU AMAÇ PROGRAMLAMA

Etkileşimli çoklu amaç programlama (EÇAP),amaç programlamanın fazla önbilgi gerektirmesi sakıncasını gidermek için önerilen yöntemlerden biridir ve Nijkamp ve Spronk tarafından geliştirilmiştir (Nijkamp v.d. ,1978).EÇAP çoklu amaç programlamanın tüm avantajlarını taşır . Örneğin öncelikler ele alınabilir.Ayrıca, etkileşimli süreç beklenti düzeyleri saptayarak , öncelikler belirleyerek , aşama aşama çözümler geliştirip beklenti düzeylerinde ayarlamalar yaparak uygulamaya benzer bir yol izler.Yöntem , karar vericinin tercih yapıcı hakkında diğer etkileşimli yöntemlerden daha fazla bilgi gerektirmez.Her türlü önbilgi yöntemde kullanılabilir.

EÇAP aşağıdaki aşamalardan oluşmaktadır:

Aşama 0 . Önce $g_i(X)$, $(i=1,........,m)$ amaç değişkenleri , $X=(x_1, x_2, ...,x_n)$ ' in doğrusal ya da parçalı doğrusal fonksiyonu olarak belirlenir. Daha sonra içinde eniyi çözümün bulunacağı olurlu bir R seti saptanır. R setinin konveks olduğu varsayılacaktır.

Aşama 1. m amaç değişkeni $g_i(X)$ 'i teker teker enbüyükleyerek eniyi çözüm g_i^* ve buna karşı gelen m değişken kombinasyon x_i^*, $(i=1,.......,m)$ belirlenir.
g_i^* 'den büyük $g_i(X)$ değeri bulmak olanaksızdır.Öte yandan

$$g_i^{min} = \min_{j=1}^{m} \{ g_i(x_j^*) \}$$

olanak tanımlandığında , g_i^{min} 'den daha düşük bir $g_i(X)$ değerini kabul etmek anlamsızdır. Zira , g_i^{min} amaç değişkenlerinin sıra ile enbüyüklemesinden ortaya çıkan en küçük $g_i(X)$ değeridir. EÇAP ' da çözüm vektörü S , sıra ile gerçekleştirilen çözümlerin içinde her amaç değişkeninin aldığı en düşük değerler olarak tanımlanacaktır.

$$\dot{I} = [g_1^*, g_2^*, ...,g_m^*]$$

$$Q = \{ g_1^{min}, g_2^{min}, ..., g_m^{min} \}$$

olarak tanımlandığında en son optimal çözüm S^*, I ideal çözüm seti ile arasında gerçekleşecektir. Ideal ve kötümser çözümler (2xm) 'lık P durum matriksi ile gösterilebilir.

Aşama 2 . Her amaç değişkeni $g_i (X)$ için karar verici aşağıdaki özelliklere sahip beklenti düzeyleri belirlemiş olabilir.

$$g_i^{min} < g_{i2} < g_{i3} ....< g_{iki-1} < g_i^*$$

$$g_{i1} = g_i^{min} \quad \text{ve} \quad g_{iki} = g_i^{*}$$

Bundan sonrakı aşamalarda beklenti düzeyleri S_i deneme çözümleri geliştirmek için kullanılacaktır. δ_j (j=1,..........,m) beklenti düzeyleri arasındaki marjinal fark olarak tanımlanır.

Karar vericinin beklenti düzeylerini önceden belirlememiş olduğu durumlarda , bu düzeyler g_i^{min} ile g_i^{*} arasında doğrusal parçalı olarak kabul edilebilir.Ek 1. de verilen bilgisayar programında bu varsayım yapılmıştır.

Aşama 3. Başlangıç çözümü

$$S_1 = \{ g_{11}, g_{21}, ..., g_{ml} \}$$

olarak tanımlanır . Bu çözüm daha önce tanımlanan kötümser çözümün aynısıdır. Çözüm P_1 ile birlikte karar vericiye sunulur.

Aşama 4. Karar verici önerilen çözüm tatmin edici ise kabul edilir . Eğer önerilen çözüm tatmin edici değilse Aşama 5. 'e geçilir. R_i , R'nin S_i amaç düzeyleri tarafından belirlenen alt seti olarak tanımlanır

Aşama 5 Karar verici şu soruya yanıtlamalıdır : Bu aşamadaki S_i çözümü verildiğinde ilk önce hangi amaç değişken iyileştirilmiştir?

Aşama 6 Karar vericinin j inci amaç değişkeni iyileştirmek istediği varsayılsın . Yeni deneme çözümü S_{i-1} gerçekleştirilir. S_{i-1} , $S_i$den yanlızca j inci değişken açısından farklıdır. Bu amaç değişkenler

$$g_j(X)_{si+1} \quad \text{ve} \quad g_j(X)_{si}$$

ile gösterildiğinde karar vericiye Aşama 2. de belirlenen bir sonraki beklenti düzeyi önerilir:

$$g_j(X)^{\wedge}_{si,1} = g_j(X)_{si} - 1/2.\delta_j$$

$g_i(X)$ in geçiçı bir değeri yukarıda tartışılan biçimde hesaplandığında

$$g_j(X) >= g_j(X)_{si+1}$$

koşulu getirilerek Aşama 7. 'ye geçirilir.

Aşama 7 Aşama 6. ya da Aşama 9. da formüle edilen kısıtlar R_j olurlu alanını tanımlayan kısıtlara eklenir. Daha sonra Aşama 1. de olduğu gibi, ancak yeni kısıtlamaları da içeren yeni bir durum matrıksi düzenlenir ve bu matriks P_{i+1} olarak isimlendirilir.

Aşama 8 Karar vericiye bir yanda S_i ve $\hat{S}_{i+1}$, diğer yanda P_i ve $\hat{P}_{i+1}$ sunulur. Durum matriksindeki farklılık, önerilen çözüme erişmek için yapılan fedakarlık olarak düşünülebilir. Karar verici bu fedakarlığı uygun bulursa, önerilen çözüm

$$S_{i+1} = \hat{S}_{i-1} \quad \text{ve} \quad P_{i+1} = \hat{P}_{i-1}$$

koyarak kabul edilir.

Karar verici fedakarlığı uygun bulmazsa, g_j (X) nin önerilen değeri gerçekten çok yüksektir. Bu durumda Aşama 7. ' de eklenen kısıtı çıkartarak Aşama 9. 'a geçilir.

Aşama 9 Karar verici açısından $g_j(X)_{si}$ nin çok düşük ve $g_i(X)_{si+1}$ ise çok yüksek olduğu bilinmektedir. Tanım gereği δ_j yi bu iki değerin farkına eşitlemek gerekmektedir. Bu durumda önerilen yeni bir S_{i+1} değeri

$$g_i(X)_{\hat{s}i+1} = g_j(X)_{si} + 1/2.\,\delta_j$$

olarak hesaplanır.

Aşama 6.' da olduğu gibi g_i(X) nin yeni önerilen değere eşit ya da büyük olmasını garantileyen kısıtlar eklenerek Aşama 7. 'ye dönülür ve durum matriksi $\hat{P}_{i+1}$ hesaplanır.

Yukarıda aşama aşama açıklanan yordama göre Basic dilinde; doğrusal programlama alt programı içeren ve Ek 1. de verilen etkileşimli bir program geliştirilmiş. Daha sonra bir yatırım sorunu program yardımıyla çözüme kavuşturulmuştur. Yatırım sorunu aşağıda açıklanmaktadır.

ÖRNEK SORUN

Bir finans şirketi , dört değişik hisse senedine yatırım planmaktadır. Şirketin üç mali analistinin bu hisse senetlerine ilişkin getiri öngörüleri Çizelge 1. de verilmiştir. Çizelgede üç analistin getiri ortalamaları , üç öngörü içinde enküçük getiri ile ortalama getiri arasındaki fark ve her hisse senedinin bu farkların ortalamasından sapmaları da çizelgede gösterilmiştir.

Hisse senetleri müdürü olan karar verici amaçlarını şöyle saptamıştır:

i. Her analistin hisse senetlerinden beklediği getirinin enbüyüklenmesi

	Hisse Senetleri			
	1	2	3	4
Öngörü 1	10	32	21	27
Öngörü 2	14	13	13	28
Öngörü 3	24	9	17	20
Ortalama Getiri	**16**	**18**	**17**	**25**
Ortalama g.- Enküçük g.	6	9	4	5
Sapma	0	3	-2	-1

Çizelge 1. Örnek Sorun

ii. (Ortalama getiri - enküçük getiri) nin ortalamasından sapma biçiminde tanımladığı riskin enküçüklemesi

Karar vericinin bu amaçları gerçekleştirirken uyması gerekli kısıtlar ise şöyledır:

i. Dört hisse senedine yapılacak yatırımın toplamı 10 milyon TL 'yi geçmemelidir.

ii. İkinci hisse senedine en az toplam yatırımın %10 'u kadar yatırım yapılmalıdır.

iii. Üçüncü hisse senedine en az toplam yatırımın %20 'si kadar yatırım yapılmalıdır.

i hisse senedine yapılacak yatırım tutarı (TL) ; x_i (i=1,2,3,4) olarak tanımlandığında bu yatırım sorunu aşağıdaki gibi ifade edilebilir:

Amaçlar :

$$\max 0{,}10\,x_1 + 0{,}32\,x_2 + 0{,}21\,x_3 + 0{,}27\,x_4$$

$$\max 0{,}14\,x_1 + 0{,}13\,x_2 + 0{,}13\,x_3 + 0{,}28\,x_4$$

$$\max 0{,}24\,x_1 + 0{,}09\,x_2 + 0{,}17\,x_3 + 0{,}20\,x_4$$

$$\min 0\,x_1 + 0{,}03\,x_2 - 0{,}02\,x_3 - 0{,}01\,x_4$$

Kısıtlar :

i. $x_1 + x_2 + x_3 + x_4 \leq 10000$

ii. $x_2 \geq 0{,}10\,(x_1 + x_2 + x_3 + x_4)$

$$-0{,}1\,x_1 + 0{,}9\,x_2 - 0{,}1\,x_3 - 0{,}1\,x_4 \geq 0$$

iii. $x_3 \geq 0{,}20\,(x_1 + x_2 + x_3 + x_4)$

$$-0{,}2x_1 - 0{,}2x_2 + 0{,}8x_3 - 0{,}2x_4 \geq 0$$

iv. $x_1, x_2, x_3, x_4 \geq 0$

Bu sorun EÇAP ile çözümlendiğinde :

Amaç Fonksiyonu	İyimser Çözüm	Kötümser çözüm
1	2980	1440
2	2350	1380
3	2110	1060
4	-150	200

başlangıç çözümüyle başlanarak kötümser değerlerden önce 4. amacın (riskin) azaltılması istenmiş, daha sonra da ikinci analistin (Hisse Senedi Müdürünün görüslerine en güvendiği analist) kötümser getiri toplamı yükseltilerek

Amaç fonksiyon	İyimser Çözüm	Kötümser çözüm
1	2630	2630
2	2350	2350
3	1830	1830
4	-80	-80

çözümü kabul edilir bulunmuştur. 1. ve 3. analistin değerlerinden ve riskin enküçük değerlerinden biraz fedakarlık edildiğinde 2. analistin en büyük değerine erişilmiştir.

SONUÇ

Görüldüğü üzere yöntemin üstünlüğü ; karar vericinin tercihlerini, diğer bir değişle iski enküçükleme amacının , getiriye enbüyükleme amaçlarına olan göreli önemini, finansal analistlerin göreli güvenilirlik derecelerini bunlara sayısal değerler atanmasına gerek kalmadan sorunun çözümünde dikkate alabilmesidir. Tercihleri daha değişik olan başka bir karar verici için çözüm de daha değişik olacaktır.

İlginç bir gözlem, bu örnek için son çözümde iyimşer ve kötümser çözümlerin yakınsamaları (convergence) olmaktadır. İyimser ve kötümser çözümlerin hangi koşullar altında yakınsama gösterdikleri ayrıca inceleme konusu yapılabilir.

AN INTERACTIVE MULTIPLE GOAL PROGRAMING APPROACH

In this paper, an iteractive approach to multiple goal programming proposed by Nijkamp et . al. (1978) is studied . The method , which is based on a successive interplay between a decision maker and an analyst with the aid of a computer; asks for the preferences and proriorities of the decision maker in a stepwise manner. Based on the approach, an interactive multiple goal program using a linear programming subroutine, is developed in Basic language. The program is then applied to an hypothetical investment decision problem.

KAYNAKÇA

Charnes A. ,W.W. Cooper;(1977) **Management Models and Industrial Applications of Linear Programming**, (1977) John Wiley and Sons , New York.

Charnes A., W.W. Copper; (1977) " Goal Programming and Multiple Objective Optimization, Part I ", **European Journal of Economic Research**, 39-54.

Moskowitz H. , G.P.Wringt ; (1979) **Operations Research Techniques for Management**, Prentice Hall Inc. , Englewood Cliffs, N.J.

Nijkomp P., J. Spronk; (1978) " Interactive Multible Goal Programming " Report 7803 /A , Centre for Research in Business Economichs, Erarmus University, Rotterdam .

Spronk, J., (1981) " Interactive Multiple Goal Programming as an Aid for Capital Budgeting and Financial Planning Under Conditions of Uncertainty", R.L. Crum, F.G.J. Derkinderen (eds.) , Martinus Nijhoff Publishing, Boston

EK 1. Bilgisayar Programı

10 REM ? copyring 1983 by Dr . Hülya Tütek

15 REM Interactive Multible Goal Programming

20 INK Q: PAPER 5 : BORDER 5: CLS : PRINT AT 1, Q; BRIGHT 1; "? INTERACTIVE M ULTIPLE ?":PRINT AT 2, Q; BRIGHT 1; "? GOAL PROGRAMMING ?": PRINT AT 3 ,Q; BRIGHT 1;"? ? COPYRIGHT 1986 ?": PRINT AT 4,Q ; BRIGHT 1; " ? BY DR . H. TUTEK ?".

25 FPR J=Q TO 31 :PRINT AT Q, J; "?":" PRINT AT 5, J; "?": NEXT J

40 INPUT "no of objektive functions : ";h

45 INPUT "no of variables : ";n: INPUT "no of constraints: " ; m

55 DIM q$(m+20): DIM s$(n) : DIM e(m+20): DIM m(m+20): DIM g (m+21,n) DIM m$(h,3>: DIM f(h,n >: DIM v(h,n): DIM b(n): DIM y(n): DIM p(h) : DIM c(n)

56 DIM d(h) : DIM u(20,h): DIM o(20,h) : DIM n(20) : DIM t (h,h+1)

57 LET a=Q: LET a,t=Q

```
60 FOR k=1 TO h
62 LET n(k)=Q
65 INPUT ( " objective function no"; k; " to be (min/max): "); LINE m$(k) : IF m$(k) ( ) " " AND m$(k) < > "max " THEN GO TO 65
68 PRINT PAPER 4 ; "OBJEKTIVE FUNCTION NO : "; k: PAPER 7: PRINT m$(k) ;" ";
70 for j=to n ; INPUT ("coefficients of objective funct #" ;k; coefficient ";j;" : ") ; x: IF j=1 THEN PRINT "(";x;")x1";
80 IF j>1 THEN PRINT "+(";X;")X";J;
85 LETf(k,j)=x: NEXT j: PRINT : NEXT k
90 PRINT : PRINT PAPER 4; "subject to"
100 FOR i=1 TO m: FOR : j=1 TO n: INPUT ( " coefficients of constraints: constraint " ;i;" ,coefficient "; j;" : ") ; x: PAPER 7 : IF j=1 THEN PRINT " ( ";x;") x1";
110 IF i>1THEN PRINT "+(";x;")x"; j;
120 LET g(i,j) = x:NEXT j
130 INPUT ("form of constraint ";i;"(,<=;=; >=):") ; LINE q$(i): IFq$(i)<>"<=" AND q$(i)<>"=" AND q$(i)<>">=" THEN GO TO 130
140 PRINT q$(i); : INPUT ("constant ";i;":") ; e(i) : NEXT i
150 FOR j=1 TO n
160 INPUT ("enter <= or >= n (for no signconstraint ); variable ";j;":") ; LINE s$(j): IF s$(j) <>"<=" AND s$(j)<>"">=" AND s$(j)<>"n" THEN GO TO 160
170 IF s$(j)<>"n" THEN PRINT "x" ; j ; s$(j0 ; "0" ;
180 NEXT j
181 INPUT "z; tocopy" ; LINE z$ : IF z$="z" THEN COPY
182 CLS
185 GOSUB 190
186 LET at=1
188 GO TO 4000
190 FOR k=1 TO h
195 FOR j=1 TO n: LET g(m+1 , j)=f(k , j) : NEXT J
200 LET to=0 : LET w=Q: FOR j=i TO n:LET t=t+ABS g(m+1, j) IF s$(j)= "n" THEN LET w=w+1
204 NEXT j: LET l=m+n+w: IF m$(k)="min" THEN LET l=2*m+n+w
207 LET t=t*10: DIM a(m+1,i+1)
210 DIM r(l): FOR i=1 TO m
220 IF m$(k)="max" THEN LET a(i,n+w+i) =1 IF q$(i)="=" THEN LET a(m+1,n+w+i)=t
230 IF m$(k)="min" THEN LET a(i,m+n+w+i)=1: LET a (i,n+w+i)=-1: IF q$(i)="=+ THEN LET a(m+1,n+w+i)=t
240 NEXT i
300 IF m$(k)="max" THEN LET g$=>="
```

```
310 IF m$(k)="min" THEN LET g$=<="
320 LET r=0: FOR j=1 TO  n: FOR i=1 TO m+1: LET a(i,j)=g(i,j)
330 IF s$ (j) ="<=" THEN LET a(i,j)=-a(i,j)
340 NEXT i:IF s$(j)="n" THEN LET r=r+1: FOR i=1 TO LET a
(i,n+r) =-1*a(i,j):NEXT i
350 NEXT J
360 FOR i=1 TO m: LET a(i,l+1) =e(i)
370 IF m$(k) ="min" AND q$(i)= g$ THEN LET a(i,n+w+i)=1
380 IF m$(k) ="max" AND q$(i)= g$ THEN LET a(i,n+w+i)=-1
390 NEXT i
410 FOR j=1 TO l+1
420 IFj>m+n+w AND j<l+1 THEN  go to 450
430 IF m$(k)= "max" THEN LET a(m+1,j) =-a(m+1,j)
440 LET z=Q: FOR i=1 TO m: LET z=z+a(i,j): NEXT i: LET a
(m+1,j)=a(m+1,j)-t*z
450 NEXT j
500 REM ** simplex algorithm**
510 LET cm=a(m+1): LET pc=1: FOR j=2 TO l: LET u=a(m+1,j)
520 IF u<cm THEN LET cm=u: LET u: LET pc=j
540 NEXT j
550 IF cm>=Q THEN GO TO 620
560 FOR i=1 TO m: IF a<i,pc>>0 THEN GO TO 630
570 NEXTj
580 PRINT : PRINT :  BEEP . 2,12: PRINT FLASH  1; PAPER 2;
"problem is unbounded": IF at< > 0 THEN go to 2100
590 INPUT "i: to start again , s: to stop" ; LINE t$: IF t$< > "s"
AND t$ < > "i" THEN  GO TO 590
600 IF t$="s" THEN go to 9000
610 CLS: GO TO 10
620 PRINT : BEEP. 2,19: PRINT  FLASH1; PAPER 2 ; "problem
has no feasible solution": IF at< > THEN GO TO 2100
625 GO TO 590
630 LET  rm=a(i,l+1)/a(i,pc): LET pr=i: FOR i=1 TO m
640 IF a(i,pc)   <=0 THEN GO TO 660
650 IF  a(i,l+1)/a(i,pc)< rm THEN LET rm=a(i,l+1)/a(i,pc): LET
pr=i
660 NEXT i
680 LET pp=a(pr,pc) : FOR j=1 TO l+1: LET a(pr, j)= a(pr,j)/pp:
NEXT j: FOR i=1 TO m+1
690 IF i=pr THEN GO TO 710
700 LET bb=-1*a(i,pc): FOR j=1 TO l+1: LET a(i,j)=bb*a(pr,j)+a
(i,j): NEXT j
710 NEXT i
720 FOR j=1 TO l:IF a(m+1,j) < 0 THEN GO TO 500
```

```
730 NEXT j
1000 LET cd=0: LET uu=Q: FOR j=1 TO l: LET ff=m+1: LET kr=0
1010 FOR i=1 TO m+1: LET kr=kr+ABS a(i,j): IF a(i,j)=1 THEN LET ff=i
1020 NEXT i: IF kr<1.0001 AND kr>.9999 AND ff<=m THEN LET r(j)=INT (a(ff,l+1)*1e5+.5)/1e5
1025 IF kr> 1+1e-5 THEN LET r(j)=0
1030 NEXT j: LET d=Q : FOR j=1 to n: let c(j)=r(j)
1040 IF s$(j)="n" THEN LET uu=uu+1: LET c(j)= r(i)-r(n+uu)
1050 IF s$(j)="<=" THEN LET c(j) =-c(j)
1060 LET d=d+c(j)* g(m+1,j): NEXT j
1070 LET Q=a(m+1,l+1): LET kb=ABS (d/0): IF kb>1.1 OR kb<.9 THEN GO TO 620
1080 FOR i=1 TO m:IF m$(k)="min " THEN LET m(i) =-a (m+1,m+n+w+i) +t: if q$(i)="<=" THEN LET m(i) =-m(i)
1090 IF m$(k) ="max" THEN LET m(i)=a(m+1,n+w+i)-t: IFq$(i)<>"=" THEN LET m(i)=a(m+1,n+w+i)
1120 NEXT i: LET obj=INT (d*1e4+SGN d/2)/1e4
1130 PRINT : PRINT PAPER 4; "opt. solution for obj. func. # ";k;":": FOR j=1 TO n: PAPER 7: PRINT "X" ; j ; "=" ; c(j) ; " " ; : NEXT j : PRINT : PRINT PAPER 4 ; "value of obj .func. # ;k; "=" ; : PRINT obj : PAOER 5
1140 FOR j=1 TO n
1150 LET v(k,j)= c(j) : NEXT j
1160 LET  b(k)=obj
1170 NEXTk
1171 INPUT "z: to copy" ; LINE z$: IF z$="z" THEN COPY
1172 CLS
1180 RETURN
2100 PRINT : PRINT " A NEW TRIAL IS BEING MADE PLESE WAIT"
2110 LET d(m)=.5*d(m): IF q$(m)="<=" THEN LET e(m) =e (m)+.5*d(ga): GO TO 4650
2120LET e(m)-.5*d(m): GO TO 4650
4000 REM POTENCY MATRIX
4005 GO SUB 4010
4007 GO TO 4200
4010 FOR k=1 TO h: FOR i=1 TO h
4020 IF i=k THEN LET y(i)=b(i): GO TO 4070
4030 LET y(i)=0
4040 FOR    j=1 TO n
4050 LET y(i)=y(i)+y(i,j)*f(k,j)
4060 NEXT j
```

```
4070 NEXT i
4080 LET ca=1
4090 FOR i=2 TO h
4095 IF m$(k)= "min" THEN GO TO 4105
4100 IF y(i) <y( ca) THEN LET ca=i
4103 GO TO 4110
4105 IF y(i)> y(ca) THEN LET ca=i
4110 NEXT i: LET p(k)=y(ca)
4120 NEXT k
4160 RETURN
4200 GO SUB 7200
4220 FOR k=1 TO h:PRINT : PRINT TAB 5;k;TAB 12;b(k); TAB
22;p(k): NEXT k
4230 INPUT "z:tocopy"; LINE z$
4240 IF z$="z" THEN COPY
4241 CLS
4400 LET a=a+1
4410 FOR k=1 TO h: o(a,k)=b(k): LET u(a,k)=p(k): NEXT k
4500 FOR k=1 TO h
4510 LET d(k) TO =.5*(b(k)-p(k)): NEXT k
4600 INPUT " IS PROPOSED SOLUTION SATISFACTORY?
(y/n) " ; LINE t$: IF t$< > "y" AND t$< > "n" THEN GO TO 4600
4610 IF t$="y" THEN GO TO 8000
4620 INPUT " WHICH GOAL VARIABLE SHOULD BE AUG-
MENTED? ";ga IF ga>h THEN GO TO 4620
4630 IF m$( ga)= "min" THEN LET e$="<=" : GO TO 4635
4633 LET e$=">="
4635 IF n(ga)=0 THEN LET m=m+1: LET n(ga)=m: LET e(m)=
p(ga)+.5*d(ga): FOR j=1 TO n: LET g(m,j)= f(ga,j): NEXT j: LET q$
(m)=e$: GO TO 4650
4640 LET ek=n(ga): LET e(ek)=e(ek)+.5*d(ga): LET q$(m)=e$
4650 FOR k=1 TO h
4660 PRINT : PRINT PAPER 4; " OBJEKTIVE FUNCTION NO :
";k: PAPER 7: PRINT m$(k);" ";
4670 FOR j=1 TO n: IF j=1 THEN PRINT "<";f(k,j)x1;
4680 IF j>1 THEN PRINT "+(";f(f,j);")x";j;
4685 NEXT j: PRINT : NEXT k
4690 PRINT : PRINT PAPER 4;"subject to "
4700 FOR i=1 TO m:FOR j=1 TO n: PAPER 7: IF j=1 THEN
PRINT "(";g(i,j);")x1;
4710 IF j>1 THEN PRINT "+(";g(i,j); ")x" ;j;
4715 NEXT j
4720 PRINT q$(i);: PRINT e$(i);: PRINT e(i) : NEXT i
4730 FOR j=1 TO n
```

```
4740 IF s$(j)< > "n" THEN PRINT "x"; j; s$(j); "0";
4750 NEXTj
4760 GO SUB 190
4810 GO SUB 4010
4820 LET a=a+1
4830 FOR k=1 TO h:LET  o(a,k)=b(k): LET u(a,k) =p(k): NEXT k
4840 PRINT "PREVIOUS SOLUTIONS"
4850 GO SUB 7200
4860 FOR k=1 TO h
4865 PRINT TAB 5;k;TAB 12;o(a-1,k); TAB 22;u(a-1,k);: NEXT k
4870 PRINT "NEW SOLUTIONS"
4873 GO SUB 7200
4875 FOR k=1 TO h:PRINT TAB 5;k;TAB 12;o(a,k); TAB 22;u(a,k): NEXT k
4880 INPUT "DO YOU CONSIDER THE CHANGE IN FEASIBLE SOLUTIONS ACCEPTABLE? (y/n) ";LINE t$ : IF t$ < > "y" AND t$< > "n" THEN GO TO 4880
4881 INPUT "z: to copy"; LINE z$ : IF z$="z" THEN COPY
4890 IF t$="y" THEN GO TO 4600
4900 INPUT " PLEASE INDICATE WHICH GOAL VARIABLE SHOULD BE REDUCED? ";ga IF ga>h THEN GO TO 4900
4910 IF n(ga)=Q THEN LET m=m+1 : LET e(m) =p(ga) -.5*d(ga) : FOR j=1 TO n:LET g(m,j)=f(ga,j): NEXT j:LET q$[m]="<=" : LET n(ga)=m: GO TO 4650
4920 LET ek=n(ga): LET d(ga)=.5*d(ga): LET e(ek)=e(ek)-d(ga): LET q$(m)="<=": GO TO 4650
7200 PRINT : PRINT "POTENCY MATRIX"
7210 PRINT :PRINT "objektive OPTIMIST PESSIMIST"
7220 PRINT "function# SOLUTION SOLUTION"
7230 PRINT "--------   ---------   --------- "
7240 RETURN
8000 REM FINAL SLUTION
8010 FOR k=1 TO h
8020 FOR j=1 TO h
8030 LET t(k,j)=f(k,j)
8040 NEXT j
8050 LET t(k,h+1)=u(a,k)
8060 NEXT k
8070 FOR k=1 TO h
8080 LET zh=t(k,k) : FOR j=k TO h+1 : LET t(k,j) /zh:  NEXT j: FOR i=1 TO h: IF i=k THEN GO TO 8100
8090 LET sh= t(i,k)/t(k,k): FOR j=k TO h+1: LET t(i,j)=t(i,j)-sh*t(k,j): NEXT j
```

```
8100 NEXT i= NEXT k
8110 PAPER 4: PRINT " SOLUTION" : PAPER 7: FOR I=1 TO
H: PRINT "x" ; i; "=";INT (t(i,n+1)*1e5+.5) /1e5: NEXT i: PAPER 6
8120 INPUT "ENTER z TO COPY";LINE l$
8130 IF l$= "z" THEN COPY
8150 SAVE "eklif " LINE 1: GO TO 8150
```