あれもこれもよくしたい多目的計画法

中山　弘隆

1.　はじめに

次の架空の話から始めましょう.『昔あるところですばらしい道具が開発され，それを用いると非常に良い結果が得られました．ところが，それを使うにはちょっとしたコツがいって，誰でもが楽に使えるというものではありませんでした．それを使いこなせる人はスペシャリストとして人々から一目おかれる存在でした．しかし，オタク族の中では少しずつ広まっていき，一方，その道具の切れ味もますます鋭く改善されていきました．そこで，一般の人々もその道具を使ってみたく思うようになりましたが，使い勝手の方は依然として悪く，そしてまたその道具をとくに使わなくても何とかやっていけましたから，やはり一般に普及するには至りませんでした.』

大学の数理計画の教育は数学的な手法ばかりでどのように使うかについて注意が払われていないという批判をときどき耳にします．数理計画はまずモデルを作ることから始まるのですが，多くの講義はモデルができたものとして，その解をどのようにして求めるかに力点が置かれているように思われます．モデルが決まればその時点で自動的に解は決まってしまうわけですから，意思決定という観点からみれば解法よりもモデル作りの方が重要になります．数理計画を有効に使うためのちょっとしたコツは最適化の手法の周辺にありますが，実際の見地からすればむしろこの周辺にあると思われていたものが案外と中心的存在だったのかもしれません．

たとえば，モデルの作り方によっては実行可能解がないという状況が出てきます．専門家にいわせるとそれはモデル作りが悪いからだと軽く一蹴されます．ところが，実際の現場で使おうとしている人に聞きますとこのような状況は結構多くあるそうです．使い込んだモデルであればどこをどのように直せばよいか大体は分かるのですが，新しい問題だとなかなか分からないこともあります．このようなとき，数理計画のソフトの方でどこをどのように直せばよいかサジェスションしてくれるものがあれば，一般の人々にとってはとても使いやすくなります．道具をうまく使いこなすちょっとしたコツをユーザーの方に求めるよりも，ソフトの方で一般のユーザーが使いやすいようにその「ちょっとした所」を工夫することも重要なことだと思います．冒頭の昔話の結末にならないようにソフト開発で気をつけなければならないことだと思います．

本稿で解説する多目的計画法もこの「ちょっとした工夫」の一つで，意思決定者の価値観をどのように数理計画モデルの中に組み入れるかを重要視しています．人間の主観的価値観を考察の対象とするところがこの特集号の他の先生方の解説とは異なっているところです．

2.　あれも，これもよくしたい

もともと人間というのは欲深いもので，あれもよくしたい，これもよくしたいと思うのが普通です．これをそのまま数理計画としてモデル化しますと，複数の目的関数をもつ問題として定式化されます．これが，多目的計画法です．通常の数理計画は，いくつかある目的のうち一つだけを目的関数にとり，他の目的は「これ以上は達成したい」という制約の形で組み込んだものと解釈することもできます．

ところで，この「他の」目的を「これ以上は達成したい」という制約の形で数理計画のモデルにいれますと，元来，人間は欲深いものですからその達成したいと思う水準がしばしば厳しすぎて，その数理計画モデルは実行不可能という事態が起こります．現在では，このようなときにどのようにすればよいかサジェスションしてくれる数理計画ソフトも出てきているようですが，1961年という昔に，この事態を初めから回避するための数理計画を考えたのが，Charnes-Cooper のゴールプログラミングです．

そもそも，数理計画生みの親ともいえるDantzigが最適化として数理計画のモデル化を行った背景にはそ

なかやま　ひろたか　甲南大学理学部
〒658 神戸市東灘区岡本 8-9-1

れまでの経済学を支配していた功利主義の思想があったものと思われます．ところが1950年代末Simonは，人間行動の合理性は最適化にあるのではなく満足化にあると主張しました．実際，多くの現場，とくに工学設計などではある程度満足できる水準にあればそれでよいとすることが多くあります．しかし満足化としてモデル化しても，先に述べたように満足できる水準が厳しすぎる場合もしばしば起こり，これを避けるためにCharnes-Cooperは「できる限り目標達成をする」という概念を導入しました．このために，いくつもある目的の目標値までの乖離をそれぞれ適当なスケールで測り，これらの総和を最小にするという最適化問題に帰着させたのがゴールプログラミングです．工学設計など多くの場でよく行われる最小2乗法は目標値までの乖離をl_2ノルムで測ったものです．すべての評価関数が線形の場合にはl_1ノルムやl_∞ノルムを使えばLPの問題に帰着されます．

目標値までの乖離を最小にするというのはLPに対するシンプレクス法における2段解法の第1段目と同じことです．乖離の総和が0になれば満足解(LPでは実行可能解)が見つかったということになります．「何だ，両者とも同じではないか」と思われるかもしれませんが，そもそものモデル化の発想が違いますから，出てきた結果の解釈が異なります．最適化としての数理計画では制約とは「何よりも前に目標値が達成されていなければならない目的」と考えることができます．ですから，そのような目標値が厳しすぎるときに古いソフトなんかを使いますと「実行可能解がありません」というメッセージが出てきて，上司からはお前のモデル作りが悪いせいだと怒られることになります．ゴールプログラミングではできる限りすべての目標を達成するというのがねらいでしたから，すべての目標値を達成してはいない(つまり，満足解が存在しない)ときにも，目標値に最も近い近似解として，出てきた解を許容することになります．もっとも，いくつもある目的間のバランスをとることが大事ですから，出てきた解に対してそのようなバランスがとれていない場合にはそのための別の方策を考えることになります．

3. あちらを立てればこちらが立たず

あれもよくしたい，これもよくしたいと思ってはいても，あちらを立てればこちらが立たずといった状況になるのが世の常です．このようなとき，あちらをどの程度立て，こちらをどの程度犠牲にするかを考えなければなりません．面倒だからエイヤッとやってしまえということもありますが，たいていの場合にはこの微妙なサジ加減に頭を悩ますことになります．このように目的間のバランスを図ることを**トレードオフ分析**といいます．実際の意思決定ではこの問題が最も重要であるといっても過言ではないでしょう．

通常の数理計画では無事に実行可能解があって，かつ最適解が出てきたとき，制約の一部あるいは全てを緩めたり，厳しくしたりして評価関数の値がそれぞれどのように変わるかを見るという感度解析(あるいは，post optimality analysis)がこれに対応します．(post optimality analysisという言葉が示すように数理計画では最適解を求めることが主で，このような分析をすることは従であるという風に一般には思われていますが，本当にそうでしょうか)．通常の数理計画における感度解析は目的関数と制約関数の感度(トレードオフ)をみるもので，いくつもある目的間のトレードオフ関係を同時に見るものではありません．多目的計画法はまさにこれを実現しようとするものです．

4. 誤った常識＝線形加重和

複数の目的関数があっても，それらに適当に重みをつけて加え合わせてそれを新たに目的関数とすれば普通の最適化問題に帰着されるではないか，と思われるかもしれません．つまり，目的関数$(f_1,\dots,f_r)$に対し

$$F = w_1f_1 + \dots + w_rf_r \tag{1}$$

というスカラー化を行うわけです．実際，このように処理している場合は非常に多いと思います．ところが，常識とも思われるこの線形加重和には思いがけない落とし穴があるのです．

【例1】 いま、目的関数がf_1, f_2と2つあってそれらをともに最小化したい場合を考えましょう．3つの代替案A, B, Cの分布が図1のようになったとします．どちらの目的にもかたよることなく，比較的平等に2つの目的を達成しているC案が意思決定者の価

値判断に照らし最も良いという場合を考えます．このとき，どのように重みw_iを調整しても(1)式の最小化ではCを解として得ることはできません．同様のことは，実行可能集合が凸多面体のときにも，LPにおけるシンプレクス法を用いる限り端点しか解として出せませんから，2つの端点の中間の部分は解として求めることはできません．

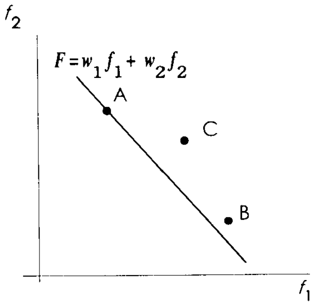

図1．線形加重和でうまくいかない例

もっと深刻なのは次の例です．

【例2】 (1)式を最小(大)化して得られた解が納得のいかない場合，通常はより改善したいと思う目的関数の重みを大きくすればよいと思われています．しかし，これが誤った常識であることは次の例を見れば明らかでしょう．

$$(f_1(x)=x_1,\ f_2(x)=x_2,\ f_3(x)=x_3)\ \rightarrow\ \text{Min}$$

$$\text{subject to}$$

$$\{(x_1,x_2,x_3)|\ (x_1-1))^2+(x_2-1)^2+(x_3-1)^2\leq 1\}.$$

$y_1=f_1(x)$, $y_2=f_2(x)$, $y_3=f_3(x)$として目的空間で考えます．最初，各目的を均等に考え，$w_1=w_2=w_3=1$にしたとします．このとき(1)式を最小化して得られる解は$(y_1,\ y_2,\ y_3)=(1-1/\sqrt{3},\ 1-1/\sqrt{3},\ 1-1/\sqrt{3})$です．さて，この解を見て意思決定者は$f_1$をもっとよくしたい，そして$f_2$をもう少しだけよくしたいと思ったとしましょう．そこで，重みを$w'_1=10$, $w'_2=2$, $w'_3=1$に変更しました．この新しい重みに対する解は$(1-10/\sqrt{105},\ 1-2/\sqrt{105},\ 1-1/\sqrt{105})$となります．$f_2$の値をよくしたいと思い対応する重みを前の2倍にしたにもかかわらず，得られた解は前よりも悪くなってしまいました．

上のようなことが起こったのはなぜでしょうか．その原因を重みの正規化を行わなかったということにあると考える人がいるかもしれません．そこで，$w_1+w_2+w_3=1$なるように重みを正規化することにしましょう．最初の重みは正規化すると$w_1=w_2=w_3=1/3$であり，解をよくしようとして修正した2回目の重みは正規化すると$w'_1=10/13$, $w'_2=2/13$, $w'_3=1/13$となります．確かに2回目の重みw'_2は1回目のw_2よりも小さくなっています．はたして，これが原因だったのでしょうか．そこで，正規化された重みw'_2が1/3よりも大きくなるようにしてみましょう．たとえば，正規化する前の重みを$w_1=10$, $w_2=7$, $w_3=1$とすればよろしい．この新しい重みに対して解は$(1-10/\sqrt{150},\ 1-7/\sqrt{150},\ 1-1/\sqrt{150})$となります．正規化された重み$w''_2=7/18$は最初の$w_2=1/3$より大きくなっているにもかかわらず，得られた解は依然として最初のものよりも悪くなっているのです．

このことから，思い通りの解の改善ができなかったのは重みの正規化を行わなかったことに原因があるのではないということが分かりました．本当の原因は重みと(1)式を最小(大)化して得られる解の間には単調な関係がないということにあるのです．このように，線形加重和のスカラー化を行い，重みの調整によって望み通りの解を得ようとするのは一般には困難であるといえます．新製品の開発設計などでは非常に多くの目的関数が考えられることがあります．これまで通常，(1)式のような線形加重和を用いて解を求めようとしている現場が非常に多いのですが，上の例のような現象が起こってなかなかうまく適切な重みを決めることができず，何度も試行錯誤をしていたずらに時間を浪費してしまうことが多くあります．しびれを切らしてエイヤッと乱暴に決めてしまうこともあります．こんな風にして出てきた解がいくらそのモデルに対して最適解だからといって，通常は意思決定の解として認めてもらえるものではありません．最適化の解法は最適解を求めるための道具ですが，モデル作りは解を決めることにつながります．意思決定者の望む解が得られるようにモデル作りをすることが最適解を求めること以上に重要だと思います．

5. 希求水準法

Charnes-Cooperによるゴールプログラミングのもつ大きな欠点は，1)複数の目標を(1)式によってスカラー化し，重みを調整することによって意思決定者が望ましいと思う解を得ようとしたこと，2)得られた解にPareto最適性が保証されない，ことだといわれています．ここで，Pareto最適とはいくつもある目的

をもうこれ以上は同時に改善できないギリギリの状態というもので，図2では太線の部分にあたります．

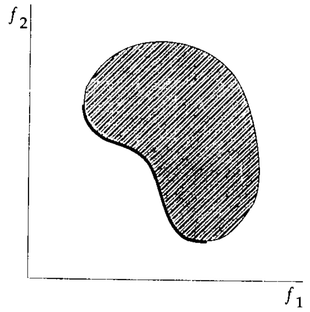

図2. Pareto 最適 (f_1, $f_2 \to$ Min の場合)

もともとゴールプログラミングは満足化を目指したものですから，解の Pareto 最適性は要求しておりません．しかし，いくつもの目的関数があったとしても数理計画の問題としてモデル化できれば，適当なスカラー化関数を用いることにより Pareto 最適な状態にまでもっていくことは容易なことです．望ましいスカラー化関数の形状や性質についてはいくつか数学的な議論がなされています．詳しいことは文献 [2] を参考にして下さい．現在最もよく用いられているスカラー化関数は次の拡大 Tchebyshev 関数です：

$$\max_{1\leq i\leq r} w_i f_i(x) + \alpha \sum_{i=1}^{r} w_i f_i(x). \qquad (2)$$

目的関数の空間で (2) 式の第2項のみの場合の等高線を描くと直線になり，第1項のみの場合の等高線は (逆)L 字型になります．したがって，(2) 式の関数の等高線はこれらの中間になり，これを伏見他 [1] はオープンL 字型と呼んでいます．この関数を使うと第4節で述べた線形加重和の問題点 1) は克服でき，適当に重み w_iを選べば任意の Pa reto 解を (2) 式の最小化によって得られるようにすることができます．ここで，通常，α としては十分に小さい値 (たとえば，10^{-6}) がとられます．

では，具体的に重み w_iはどのように定めればよいのでしょうか．1つのアイデアは (2) 式を用いるときに各項がそれなりの意味をもつようにすることです．たとえば，満足化トレードオフ法 [2] では (2) 式を若干変更して次のようなスカラー化関数を用います．

$$\max_{1\leq i\leq r} w_i(f_i(x) - \overline{f}_i) + \alpha \sum_{i=1}^{r} w_i f_i(x). \qquad (3)$$

ここで，$\overline{f_i}$は第 i 番目の目的関数に対する希求水準 (この程度あれば，望ましいという水準) です．さて，第 i 番目の目的関数に対する理想値を f_i^* とします．これは，たとえば f_iの値が小さいほどよいときには

$$f_i^* < \min\{f_i(x) | \ x \in X\}$$

で定めてもよいし，経験上分かっていればそれを用いてもよろしい．このとき，w_i を

$$w_i = \frac{1}{\overline{f}_i - f_i^*} \qquad (4)$$

とすることができます．このようにすれば，(3) 式の第1項は理想点 f_i^*を基準として測った希求水準$\overline{f_i}$までの不達成度とみなすことができます．ここで，マイナス値をとることを許していますから，(3) 式を最小にするということは希求水準を達成してもなおかつできる限り f_iの値が小さくなるように努力することになるということに注意して下さい．これが Pareto 最適性を保証する大きな原因となります．また，伏見他の目標ベクトル法でも類似の設定がなされています．

このように，各目的関数を理想点と希求水準をもとにしたスケーリングによって無次元化したそれぞれの目的の不達成度をできる限り平等に最小化しようとしたものが (3) 式であると解釈できます．一応，このような設定の仕方に大きな異論はないと思います．ところで，このようにすれば，意思決定者の望む解は本当に得られるのでしょうか．実際，(4) 式を用いた (3) 式の最小化によって得られた解は，希求水準にある意味で最も近い Pareto 解ですから，大体は意思決定者に満足してもらえそうです．しかし，当初の希求水準が達成不可能であった場合などではとくに，得られた解の不達成度のアンバランスに不満が生じることがあります．この目的がこの程度の値であれば，もう少し犠牲にしてもかまわないから別の目的をもっと改善したい，と思うことがしばしばあります．先にも述べましたように，数理計画では出てきた解にただちに納得できるとは限りません．あるモデルに対してどのような解になったかを知った時点で，新たな情報を得たわけですから，意思決定者は態度を変えることもあります．たとえ，意思決定者の価値観が変わったとしても，新しい価値観に即応した解を求めるために，何を探索の手がかりにすればよいでしょうか．意思決定者の価値観は希求水準に反映されていますから，それを直接に手がかりにして解の探索を行うのが最も望ましいと考えられます．このようにして，希求水準法が考えられました．

6. トレードオフ分析

希求水準法では意思決定者の望み通りの解が得られるまで，希求水準を変更します．アルゴリズムはおおまかにいって次の2つのステップからなりたちます．

Step 1. (補助最適化) 与えられた希求水準に対し，何らかの意味で最も近いPareto最適な解を求め意思決定者に提示します．このための1つの方法が(4)式で与えた重みをもつ(3)式を最小化することであることは先に述べました．(3)式の最小化はそのままでは滑らかでない目的関数をもつ最適化問題になるので，通常は次のように等価な問題に変換して解きます．

(Q)　Minimize　$z + \alpha \sum_{i=1}^{r} w_i f_i(x)$

subject to

$$w_i(f_i(x) - \overline{f}_i) \le z \quad (i = 1, \ldots, r) \tag{5}$$

$$x \in X.$$

Step 2. (トレードオフ)　さて，示された解に納得できないとき，全体のバランスを考え，意思決定者は希求水準を設定し直します．現在の解はすでにPareto最適ですから，ある目的を改善しようと思えば，別の目的を犠牲にせざるをえません．また，現在の解はある程度最初の価値判断に沿ったものですから，その解に納得できないといっても，普通は新たな希望は現在の解から極端に離れるということはありません．あちらをどの程度たて，こちらをどの程度犠牲にするかというトレードオフ分析は現在の解を基準にしてそこからどう動くかを判断すればよいと考えられます．

さて，このトレードオフ分析ですが，これは意思決定者の主観的価値判断の問題ですから意思決定者によって変わるものです．このようにユーザーによって変わるものを最初からシステムの中に組み込んでモデル化して最適化することはほとんど不可能です．したがって，補助最適化によって示された解を見て，納得できなければ意思決定者に再度希求水準を入れ直してもらうという対話型解法をとります．ここで，改善したいと思う目的の新しい希求水準とともに，犠牲にしてもよいと思う目的の新しい希求水準を答えてもらっても，もちろん構いません．しかし，問題によっては何百という目的がある場合もあって，このようなときには全ての目的に対して新しい希求水準を答えてもらうということは意思決定者にとって負担が大きすぎます．しかも，意思決定者は通常は欲張りで，犠牲にすべき目的に対し，なかなか思い切った譲歩はしないものです．犠牲にすべき量が不十分ですと，改善したいと思った目的の値が達成されません．

そこで，意思決定者には改善したいと思う目的の新しい希求水準のみを答えてもらい，あとはほんの少し数理計画の数学的性質を使って，犠牲にすべき量の目安を求める方法が考えられました．Lagrange乗数の情報によって，Pareto曲面の線形近似上に更新した希求水準がのるようにすることです(満足化トレードオフ法における自動トレードオフ)．さらに，もとの問題が線形計画あるいは2次計画の形をしていれば，正確に譲歩すべき量を割り出すことができます(満足化トレードオフ法における適正トレードオフ)．

これらの計算はコンピュータを使えばあっという間にできますから，意思決定者は改善したいと思う目的の新たな希求水準を入力した直後に，ほぼ瞬間的に，他の目的の犠牲にすべき量を知ることができます．犠牲にすべき量が思いもかけず大きすぎれば，改善すべき量をもう少し控えめにしておこうと考えるようになります．逆に，思ったほど犠牲を強いる必要がないのであれば，改善すべき量をもっと多くしようということになります．この段階でこのようにある程度トレードオフ分析をしておけば，新しく設定し直した希求水準に対して(3)式の最小化をしなおしたとき，ほとんど設定した希求水準に近いPareto解が得られますから，意思決定者も出てきた解に満足しやすくなります．

以上のようなトレードオフ分析の仕方をみますと，通常の数理計画における感度解析と同じではないかと思われるかもしれません．実は，技法的にはその通りなのです．しかし，通常の数理計画では，制約関数の目的関数に対する感度(トレードオフ)をみるのに対し，多目的計画法では目的間相互のトレードオフ分析を行いますからもっと直接的に総合的なバランスを図ることができます．制約の右辺値が変わり得るものは，むしろ最初から目的関数とみなし，多目的計画法としてモデル化し，意思決定者の価値観に対して開いたシステムとして対話型解法を用いた方が意思決定支援の道具としては柔軟性が出てくると思います．

さらに，解の探索のプロセスの中で，ある目的に対してはどうしてもこの水準は確保したいと思うこともあります．このとき，この目的はハードな制約とすべきなのですが，補助的スカラー化最適化問題(Q)において，(5)式の右辺にパラメータβ_iを導入することに

よって自由に目的と制約の役割を入れ換えることができます．つまり

$$w_i(f_i(x)-\overline{f}_i) \leq \beta_i z. \qquad (6)$$

として，$\beta_i=1$とすれば，関数f_iは目的関数として取り扱われ，可能な限りその値が小さくなるようにされますが，希求水準は必ずしも達成される必要はありません．また，$\beta_i=0$のときには，f_iは制約関数として取り扱われ，希求水準$\overline{f}_i$は必ず達成されることが要請されます．さらに，β_iの値を0から1まで動かすことにより，完全な制約よりは目的に近いもの，あるいは，完全な目的よりは制約に近いもの，というように状況によって意思決定者の思いのままに扱うことができます．

《注》　希求水準を変えることによって望ましい解の探索を行っているといっても，(4)式をみれば結局は重みを変えていることになるではないか，と思われるかもしれません．実際，その通りです．第4節で述べた問題点は線形加重和のスカラー化をとって，かつ重みの修正によって解の探索を行うという取り合わせにあるのです．線形加重和ではなく，たとえば(3)式のようなスカラー化を行えば，解の探索に重み修正という方法をとっても第4節で述べたような問題点はある程度克服できます．しかし，重みはその結果得られる解に定量的な関係を直接にはもちませんから，重み修正による方法ではなかなかうまく意思決定者が望ましいと思う解にいきつくことができません．意思決定者の価値観を直接的に反映している希求水準を手がかりにした方が，意思決定者にまどろこしさを感じさせないですみます．

それでは，第4節で扱った例題をここでもう一度希求水準を手がかりにして解いてみましょう．理想点は$f^*=(0,\ 0,\ 0)$となります．最初の希求水準が(0.4, 0.4, 0.4)であったとしますと，補助的最適化(Q)を解いて得られる解は(0.423, 0.423, 0.423)となります．この解を見て意思決定者はf_1の値をもっとよくしたい，f_2の値をもう少しよくしたいと考えたとしましょう．そこで，新しい希求水準として$\overline{f_1}=0.35$, $\overline{f_2}=0.4$にしたとします．自動トレードオフ法を用いますと，残りの目的f_3は0.52まで犠牲にしなければならないということが，線形近似としてですが，出てきます．この新しい希求水準がOKですと，これに対する補助最適化を行い，新たなPareto解(0.354, 0.404, 0.524)を得ます．新希求水準にまでは至りませんでしたが，意思決定者の思惑通りになっております．

7. むすび

数理計画を用いる際，モデル作りが大事なことはよく知られたことです．普通，このモデル作りといったときには目的関数や制約関数の関数形を求めることと思われていますが，何を目的関数にとるか，また何を制約関数にとるか，を決めること自体重要なことです．多目的計画法を用いれば，変えることのできないハードな制約だけを制約として取り扱い，もともと目的と考えられるものはもちろん，右辺値が変えられるソフト制約も最初の段階で目的関数として取り扱い，決定プロセスの途中で適宜必要に応じてその役割を変えることができることを知りました．また，モデル作りの本来の役割はそれによって解を定めることにあります．意思決定者の望み通りの解が得られるように工夫する必要がありますが，最初からそのようなモデル作りを行うことはほとんど不可能です．意思決定者の価値観をとりこみながら解を探索するという対話型のモデルを解説してきました．このように，柔軟に数理計画を適用できるようにヒューマンインターフェイスを工夫することが多目的計画法のつとめでもあります．どのようなタイプの最適化であれ，そのためのソフトウェアがありさえすれば，それをエンジンとして多目的計画法のソフトを開発することができます．具体的なヒューマンインターフェイスは個々の問題に応じて，工夫を凝らすべきものですが，一般的な利用に供するものとして，いくつかの有償，無償のソフトウェアもすでに開発されています．詳しくは，文献[2]を参考にして下さい．

最後に，冒頭の架空の話が次のような実話になることを祈って筆を置きます．『昔あるところで数理計画というすばらしい意思決定支援のための道具が開発され，それを用いると非常に良い結果が得られました．その後，その切れ味もますます鋭く改善されていき，また，使い勝手の方も工夫され，実際の場でも広く普及するようになりました．』

参考文献

[1] 伏見多美雄，福川忠昭，山口俊和："経営の多目標計画"，森北出版(1987).

[2] 中山弘隆，谷野哲三："多目的計画法の理論と応用"，計測自動制御学会(1994).