特集　ＩＡ法

数理計画法の立場から見たIA法

伊　理　正　夫

ＩＡ法 (incremental assignment method) というものの定義も評価も世界的にまだ確立されているとはとてもいえないが，その特徴を正しく把握して実際問題に上手に応用しているという実績では，わが国はかなりの先進国であるといってよかろう．本稿でこれから述べるように，“数理計画法”的観点から眺めた場合，ＩＡ法はとくに奇抜な着想を含んでいるわけではなく，ごく地味な方法である．しかし，本特集で奥平氏が解説しておられるように[1]，ＩＡ法は，

(i) 実際の現象をかなり忠実に模擬しているとも解釈できるような算法を用いており，

したがって，

(ii) 最終的に収束した結果だけでなく，中間結果も十分な利用価値をもち，

また，計算機を使用する立場からは，

(iii) 必要な記憶場所が最小ですむ；すなわち，問題を記述する場所だけがあればたいていの場合十分である（多くの場合，数理計画問題として定式化したときの“全変数”を記録する場所すら必要としない）

ので，

(iv) かなり大きな規模の問題まで小さな計算機で処理することができるし，

(v) プログラムづくりも容易である（一応は！）

というような多くの特長を備えている．もちろんこれらの特長に対して欠点もあるわけで，

(vi) 収束速度は遅く，

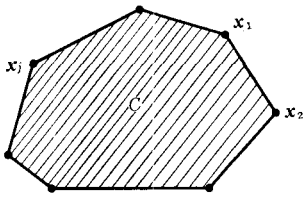
(vii) 得られる解の精度も高くない．

これらの欠点を少しでも減らそうという工夫も問題の型に応じていろいろなされてはいる．

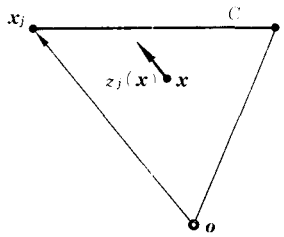
1. 数理計画法による問題の定式化——原型

N 次元実ユークリッド空間 R^N の中の（原点 $\boldsymbol{o}$ を含まない）有界閉凸多面体 C の上で凸関数 $z(\boldsymbol{x})$ の値を最小にすることが問題であるとする．C の頂点を $\boldsymbol{x}_j(j=1, 2, \cdots)$ と記す．また，z は，$C\cup\{\boldsymbol{o}\}$ の凸閉苞 $\overline{C\cup\{\boldsymbol{o}\}}$ を含むある開集合で定義された連続微分可能な凸関数で，その導関数 ∇z（ベクトル）は Lipschitz 連続（Lipschitz 定数は L）であるとする．

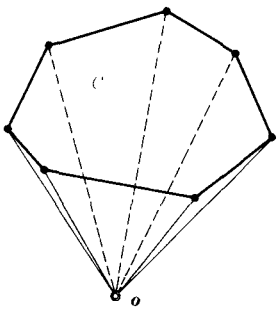
$\overline{C\cup\{\boldsymbol{o}\}}$ 内の任意の点 $\boldsymbol{x}$ における関数 z の“頂点



(a) 凸多面体 C とその頂点



(c) 方向微分 $z_j(\boldsymbol{x})$



(b) $\overline{C\cup\{\boldsymbol{o}\}}$

図 1　基本的概念の定義

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

x_j 方向への方向微分" $z_j(x)$ (スカラー)を

$$z_j(x)=x_j\cdot\nabla z \qquad (1)$$

により定義する(図 1).

以上の諸定義のもとに，ＩＡ法の算法を述べるとつぎのようになる.

〈Phase Ｉ〉 Nはあらかじめ定めた正整数である．初期解は $x^{(0)}=o$ とおく．$m=0, 1, \cdots, N-1$ に対して，$x^{(m)}$における方向微分 $z_j(x^{(m)})$がもっとも小さいようなCの頂点x_{j_m}を選び，

$$x^{(m+1)}=x^{(m)}+\frac{1}{N}x_{j_m} \qquad (2)$$

とおく．$x^{(N)}$($\in C$となることは後で示す)がPhase Ｉにより得られた解である.

〈Phase Ⅱ〉 Cの中の任意の点(Phase Ｉで得られた解を用いるのが普通である) を初期解 $x^{(0)}$とする．ε をあらかじめ与えられた"小さな"正数とする．$m=0, 1, 2, \cdots$(無限に繰返す)に対して，$x^{(m)}$における方向微分 $z_j(x^{(m)})$がもっとも小さいようなCの頂点x_{j_m}を選び，

$$x^{(m+1)}=(1-\varepsilon)x^{(m)}+\varepsilon x_{j_m} \qquad (3)$$

とおく.

図2は上の算法を直観的に説明するものである．まず，Phase Ｉでは，われわれは原点oから出発して，oと実行可能領域Cとの間をN等分し，NステップでC内の１点に到達しようとする．各ステップごとに，前進する方向は原点とCの一つの頂点を結ぶ方向に平行な方向とするが，それらの可能な方向の中ではそれにそっての $z(x)$ の増加率が最小であるものを選ぶ (Cに到達したときの$z(x)$の値が，そうすれば，小さいであろうとの期待を込めて)．初期のＩＡ法はこの Phase Ｉのみで終わりにしていた．しかし，これでは問題の最適解が得られるとはかぎらないことは明らかであろう.

Phase Ⅱでは，現在いるところから原点へ向けてεだけ後退し，再びC内の点へもどるとき，もどる方向をzの増加率が最小であるような頂点の方向に一致させることによって，次第に解の改善をはかろうとする.

2. 最適解への収束性の証明

ＩＡ法によると，Phase Ｉにおいて問題の一つの実行可能解(C内の点)が求められ，Phase Ⅱにおいてそれが次第に改善されて最適解に近づいていく．もっとも，それはεが十分小さい理想的な場合であって，εがある大きさをもっている場合には最適解からは少々ずれたところに落着くことになる．このような現象をやや厳密に述べてみよう.

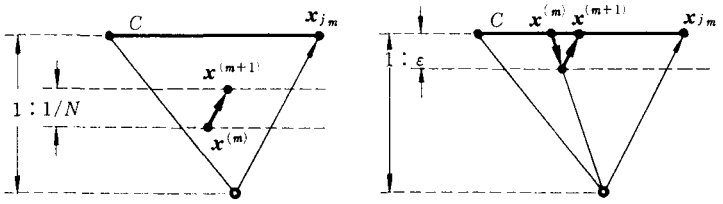
Phase Ｉで得られる解が一つの実行可能解であることは直ちにわかる．実際，$x\in C$ であることと，

$$x=\sum_j \omega_j x_j,\ \ \omega_j\geqq 0,\ \ \sum_j \omega_j=1 \qquad (4)$$

であるような"重み"の組$\{\omega_1, \omega_2, \cdots\}$が存在することとは等価であるが，(2)によれば，

$$x^{(N)}=\sum_{m=0}^{N-1}\frac{1}{N}x_{j_m} \qquad (5)$$

である．(このことからわかるように，$x^{(0)}=o$から $x^{(N)}\in C$ の間は $1/N$ ずつに"等分"されている必要はなく，各ステップごとに"$1/N$"のかわりに適当な$\nu_m(\geqq 0)$を用い，最終的に$\sum_m \nu_m=1$となるようにすればよいのである.)



(a) Phase Ｉ　　(b) Phase Ⅱ

図 2　ＩＡ法の Phase Ｉ と Phase Ⅱ

Phase Ⅱにおいて得られる解の列 $x^{(0)}, x^{(1)}, \cdots$がもとの問題($\min_{x\in C} z(x)$) の最適解に"収束する"ことはつぎのように考えれば理解できよう．まず，(3)から $x^{(m+1)}-x^{(m)}=\varepsilon(x_{j_m}-x^{(m)})$であり，また$z$の導関数が Lipschitz 連続であ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

るという仮定も考えに入れると，1ステップあたりの関数 z の値の変化は，

$$\begin{aligned} z(\boldsymbol{x}^{(m+1)})-z(\boldsymbol{x}^{(m)}) &= (\boldsymbol{x}^{(m+1)}-\boldsymbol{x}^{(m)})\cdot \\ &\quad \nabla z(\boldsymbol{x}^{(m)}+\theta_m(\boldsymbol{x}^{(m+1)}-\boldsymbol{x}^{(m)})) \\ &= \varepsilon(\boldsymbol{x}_{j_m}-\boldsymbol{x}^{(m)})\cdot\nabla z(\boldsymbol{x}^{(m)}) \\ &\quad +M_m\theta_m\varepsilon^2\|\boldsymbol{x}_{j_m}-\boldsymbol{x}^{(m)}\|^2 \\ &= \varepsilon(z_{j_m}(\boldsymbol{x}^{(m)})-\sum_j \omega_j{}^{(m)}z_j(\boldsymbol{x}^{(m)}))+E_m \end{aligned} \tag{6}$$

という形にあらわすことができる．ここで，$\omega_j{}^{(m)}$ は $\boldsymbol{x}^{(m)}$ を C の頂点 $\boldsymbol{x}_j$ の凸一次結合であらわすときの係数：

$$\boldsymbol{x}^{(m)}=\sum_j \omega_j{}^{(m)}\boldsymbol{x}_j,\quad \omega_j{}^{(m)}\geqq 0,\quad \sum_j \omega_j{}^{(m)}=1$$

であり，θ_m は $0<\theta_m<1$ なるある数，$|M_m|<L$ である．したがって，C の“直径”を

$$D=\max_{\boldsymbol{x},\boldsymbol{y}\in C}\|\boldsymbol{x}-\boldsymbol{y}\| \tag{7}$$

と記せば，(6)の E_m は

$$|E_m|\leqq D^2L\varepsilon^2 \tag{8}$$

を満たす．$\boldsymbol{x}_{j_m}$(および z_{j_m})の定義により，

$$z_{j_m}(\boldsymbol{x}^{(m)})-\sum_j \omega_j{}^{(m)}z_j(\boldsymbol{x}^{(m)})\leqq 0 \tag{9}$$

であるから，Phase IIの1ステップは，ε が十分小さければ，乱暴にいって，“改善”になっているといえる．この辺の事情をより厳密に述べるとつぎのようになる．

問題の最適解の一つを $\hat{\boldsymbol{x}}$ とし，ω_jを

$$\hat{\boldsymbol{x}}=\sum_j \hat{\omega}_j\boldsymbol{x}_j,\quad \hat{\omega}_j\geqq 0,\quad \sum_j \hat{\omega}_j=1 \tag{10}$$

で定義する．z が凸関数であるから，

$$\begin{aligned} 0 &\geqq z(\hat{\boldsymbol{x}})-z(\boldsymbol{x}^{(m)})\geqq(\hat{\boldsymbol{x}}-\boldsymbol{x}^{(m)})\cdot\nabla z(\boldsymbol{x}^{(m)}) \\ &= \sum_j \hat{\omega}_j z_j(\boldsymbol{x}^{(m)})-\sum_j \omega_j{}^{(m)}z_j(\boldsymbol{x}^{(m)})\ ; \end{aligned} \tag{11}$$

さらに，$z_{j_m}(\boldsymbol{x}^{(m)})$の定義により，

$$\begin{aligned} &z(\hat{\boldsymbol{x}})-z(\boldsymbol{x}^{(m)})\geqq \\ &\quad z_{j_m}(\boldsymbol{x}^{(m)})-\sum_j \omega_j{}^{(m)}z_j(\boldsymbol{x}^{(m)}). \end{aligned} \tag{12}$$

(12)を(6)に代入して，

$$\begin{aligned} &z(\boldsymbol{x}^{(m+1)})-z(\boldsymbol{x}^{(m)})\leqq \\ &\quad \varepsilon(z(\hat{\boldsymbol{x}})-z(\boldsymbol{x}^{(m)}))+E_m, \end{aligned}$$

あるいは，

$$\begin{aligned} &z(\boldsymbol{x}^{(m+1)})-z(\hat{\boldsymbol{x}})\leqq \\ &\quad (1-\varepsilon)(z(\boldsymbol{x}^{(m)})-z(\hat{\boldsymbol{x}}))+E_m \end{aligned} \tag{13}$$

を得る．(8)を考慮しつつ(13)を繰返し用いれば，

$$\begin{aligned} &|z(\boldsymbol{x}^{(m)})-z(\hat{\boldsymbol{x}})|\leqq \\ &\quad (1-\varepsilon)^m|z(\boldsymbol{x}^{(0)})-z(\hat{\boldsymbol{x}})|+D^2L\varepsilon \end{aligned} \tag{14}$$

という評価式を得る．すなわち，パラメタ ε を C の直径 D および z の導関数の連続率 L に比べて十分小さく選んでおけば，Phase IIを繰返しているうちに $\boldsymbol{x}^{(m)}$は——$z(\boldsymbol{x}^{(m)})$が $z(\hat{\boldsymbol{x}})$に近づくという意味で——次第に改善されていく．なお，以上の考察からわかるように，ε は各ステップごとに異なる値を用いてもよい．

3. パラメタ ε の選択についての注意

前節の議論(とくに式(14))から推察されるように，IA法を用いる際にもっとも工夫を要するのはパラメタ ε の値の選び方である．IA法は勾配法の一変種ともみなせ，収束過程としては1次過程である．収束速度を上げようとすると ε を大きく($1-\varepsilon$を小さく)選ばなければならないが，そうすると(14)における誤差項 $D^2L\varepsilon$ が大きくなってしまう．しかし，さらにくわしく $D^2L\varepsilon$ の項の由来を調べればわかるように，ε は

$$\begin{aligned} &|(\boldsymbol{x}^{(m+1)}-\boldsymbol{x}^{(m)})\cdot\nabla z(\boldsymbol{x}^{(m)})- \\ &\quad (\boldsymbol{x}^{(m+1)}-\boldsymbol{x}^{(m)})\cdot\nabla z(\boldsymbol{x}^{(m+1)})| \end{aligned} \tag{15}$$

が小さくなるように選べばよいので，したがってPhase IIの各ステップにおいて，(15)があらかじめ定められた値以下になる範囲で ε を大きく選ぶという工夫をすれば，全体としての解法の速度が上げられる．

ε の選び方を誤ると，ときには，およそ最適解とはほど遠いところへいってしまったり，あるいは収拾のつかないほどの激しい振動を起こしたりすることもある．この辺の事情については，具体例にもとづいた明快な森口先生の本特集の解説[2]をご覧いただきたい．

4. 例題による説明

IA法を上手に使いこなすためには，上に述べ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

てきたような数理計画法の“理論的”面を理解しただけでは不十分であって，計算機の記憶装置の中に蓄えておくべき情報を必要最小限にとどめ，また不必要な計算をすることがないように計算手順の設計をしなければならない．このような実際的な配慮について一般的に述べることは不可能ではないにしてもあまり有効とは思えないので，以下では一つの例について前節までにあらわれた各種の概念の物理的なイメージを説明しつつ実際的な注意を述べていくことにする．

いま，q 個の会社が共通の道路網を用いてそれぞれ1対の地点間に定常的にトラックの運行をしているとする．会社 i は単位時間あたり r_i 台を運行するが，路線は混雑状況を考慮に入れて最短時間ですむように選びたい($i=1,\cdots,q$)．道路網は l 本のリンク(各道路の片側を1本と数える)および n 個のノード(交差点など)よりなっており，これらの会社の車以外にはこの道路網を使う車はないとする(図3)．各道路(リンク)は，そこを単位時間に通過する車の数 ξ によって混雑状況が異なるので，そこを通過するのに要する時間は ξ の関数 $\varphi_\kappa(\xi)$ で与えられる（道路ごとに異なる）(図4)．ノード α から出ているリンクの集合を δ^+_α，それに入っているリンクの集合を δ^-_α と記し，会社 i のトラックの出発ノードを A_i，到着ノードを B_i と記す．

この問題を数理計画問題としてあらわすには，もっとも簡単な表現をするにしても，各リンク κ を通過する会社 i のトラックの数 $x_\kappa{}^i$ を一応は変数として考えねばならないであろう．(ここで，もしノードの数が $n=500$，リンクの数が $l=2,000$，会社の数が $q=1,000$あったとすると，変数 $x_\kappa{}^i$ の総数は $lq=2,000,000$となる．)変数 $x_\kappa{}^i$ が満足すべき条件は各ノードにおけるフローの連続条件で，それらは，各 $\alpha(=1,\cdots,n)$，各 $i(=1,\cdots,q)$ に対して，

$$\sum_{\kappa\in\delta^+_\alpha} x_\kappa{}^i-\sum_{\kappa\in\delta^-_\alpha} x_\kappa{}^i=\begin{cases}0 & (\alpha\neq A_i,B_i),\\ r_i & (\alpha=A_i),\\ -r_i & (\alpha=B_i)\end{cases}\qquad(16)$$

という形であらわされる．(16)は全部で nq 個(上記の規模のネットワークでは$=500,000$個)の等式群であり，これらと $x_\kappa{}^i\geqq 0$(すべての i, κ)を満足する R^{lq} の部分集合がいまの場合の C である．このように，C を式で表現するとかなり厄介なことにはなるが，C の頂点の物理的意味は明らかである．それは $x_\kappa{}^i>0$ であるような (κ,i) の集合が極小であるような(16)の解として特徴づけられるから，そして，(16)は各会社ごとに独立した条件式群であるから，C の頂点 $x_j=(\underset{j}{x_1{}^1},\cdots,\underset{j}{x_l{}^1};\underset{j}{x_1{}^2},\cdots,\underset{j}{x_l{}^2};\cdots\cdots;\underset{j}{x_1{}^q},\cdots,\underset{j}{x_l{}^q})$ は各会社が A_i から B_i への適当な一つの道にそって r_i だけの車をすべて割当てたときの流れの模様をあらわすベクトルであるということになる．もちろん，このような頂点をあらかじめすべて列挙しておくことは不可能であるが，後に述べるように，“ある特徴をもった頂点を見いだす”ことはこのような物理的解釈から容易であるのが普通である．

さて，各会社が“最短時間”をめざして運行計画を立てると，結局は，(16)の条件(と $x_\kappa{}^i\geqq 0$)のもとで，

$$z(x)=\sum_{\kappa=1}^{l}\int_0^{\sum_{i=1}^{q}x_\kappa{}^i}\varphi_\kappa(\xi)\,d\xi\qquad(17)$$

という関数を最小にするようなところに落着くことが知られている（Wardrop の“等時間原理”；たとえば，文献[3]参照）．

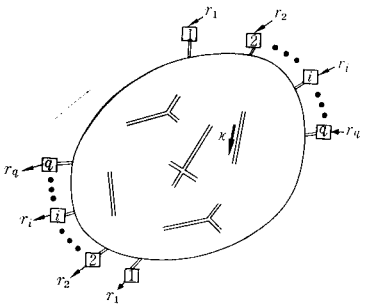


図 3　道路網と交通需要

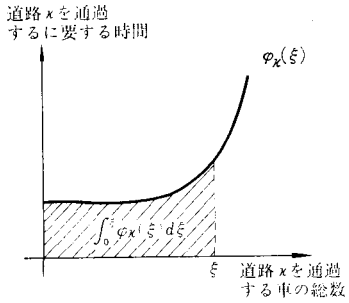


図 4　道路の特性

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

(17)の関数 $z(x)$ は $x_\kappa{}^i \geqq 0$ であるようなすべての x に対して定義されており，φ_κ が単調増大であるから $z(x)$ は凸関数である．

この形の問題を §1 で述べた形のＩＡ法で解こうとすると，どういう計算をしなければならないことになるであろうか．$x=o$（すべての道が空の状態）から出発すること—これは容易である．ある流れの状態 x において ∇z の第 $(i,\ \kappa)$ 成分は，(17)より，

$$\frac{\partial z}{\partial x_\kappa{}^i}=\varphi_\kappa\left(\sum_{i=1}^{q} x_\kappa{}^i\right) \tag{18}$$

とあらわされる．そこで，C の頂点 $x_j=(\underset{j}{x_1}{}^1,\cdots,\underset{j}{x_l}{}^1;\cdots;\underset{j}{x_1}{}^q,\cdots\underset{j}{x_l}{}^q)$ の方向への方向微分は，

$$z_j(x)=x_j\cdot\nabla z(x)=\sum_{i=1}^{q}\left[\sum_{\kappa=1}^{l}\underset{j}{x_\kappa}{}^i\varphi_\kappa\left(\sum_{i=1}^{q}x_\kappa{}^i\right)\right] \tag{19}$$

となる．(19)の[……]の中は，会社 i の出発点 A_i から到着点 B_i へのある道にそっての所要時間（各リンクの通過所要時間が，いまの混雑状況のもとでは，$\varphi_\kappa(\sum^q{}_{i=1}x_\kappa{}^i)$ である）の r_i 倍に等しい．したがって，$z_j(x)$ が最小である頂点 x_j は，

各会社ごとに現在の混雑状況のもとで最短時間の道を選んでそれにすべての車を割当てる

という形の解として定められる．このような解は，$\varphi_\kappa(\sum^q{}_{i=1}x_\kappa{}^i)$ を各リンクの“長さ”に見立てて，各会社ごとに“最短経路問題”を解くことによって得られる．x に対して $(1-\varepsilon)x$ をつくること，$+\varepsilon x_j$，$+\frac{1}{N}x_j$ をつくること，等は容易である．

われわれが lq 個の変数をすべて必要とするのならば以上の手続を実際に行なわなければならない．しかし，もし最終的には各道路の混雑状況，すなわち $\sum^q{}_{i=1}x_\kappa{}^i(\kappa=1,\cdots,\ l)$，だけに関心があるのであれば（そして，この種の問題ではこのような場合がほとんどであるが），すべての $x_\kappa{}^i$ を記録しておかなくても，各 κ ごとの $\sum^q{}_{i=1}x_\kappa{}^i$ を記録しておきさえすれば，上記の計算が遂行できることがわかるであろう．ネットワークの構造を記録し最短経路問題を効率よく解くための工夫をするために必要な場所まで含めて，実際には，$6n+9l+3q$（上の例では＝24,000）語程度の記憶場所があれば十分である．（このようにするためのデータ構造や算法についての詳細は文献［4］，［6］を参照されたい．）

§2 および §3 で述べたパラメタ ε（および ν_m）の選び方についての注意はいまの場合つぎのように述べられよう．「会社 i が選んだ最短路上のリンクの車数は，まず $(1-\varepsilon)$ 倍されてつぎに εr_i だけ増加させられるわけであるから，この変化を通じてそのリンク通過所要時間があまり大きく変わらない程度に ε を選ぶこと．」すなわち，「大きな r_i の会社が幅の狭い道路を含む最短路を選んだときには ε は小さく」ということである．

本題からははずれるが，上のように述べると各ステップごとに q（＝1,000）回最短路問題を解かなければならないようにもみえるが，実際には，出発点（あるいは到着点）を共有する会社の最短路は同時に求めることができるので，最短路問題を解く回数は $\sqrt{q}$ 回程度ですむことが多い．

ＩＡ法は，ある見方をすれば，Dantzig, Wolfe, Frank 等の分解原理を利用しているともいえる（双対変数 φ_κ を仲介にして問題を会社ごとの部分問題にわけてしまう）．そして，頂点 x_j を求めるのは一種の column generation の技巧であるとみなせる．また，ＩＡ法が非常に遅い勾配法なのでもっと加速しようという話もある．（文献［6］，［7］参照．）しかし，“大規模な問題を小さな計算機で”処理しようとするときには，収束速度と精度をあまり問題にしなければ，ＩＡ法は大変重宝な方法である．

5. 余分な制約のあつかい方

実行可能領域 C が凸多面体でその頂点 x_j が容易に求められるような制約しかない場合についてはすでに述べたとおりであるが，そのような制約以外に余分な制約がある場合にもＩＡ法を使うこ

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

とは可能である．それには，“あつかいやすい制約”は C の形であつかい，“余分な制約”は，たとえば，罰金関数(penalty function)を用いて目的関数 $z(\boldsymbol{x})$ の中に組み入れてしまえばよい．

§4の例題でいえば，各道路の通過所要時間が車の数がある限度を超えると∞になる，すなわち各道路に“容量制限”のある場合などである．このような場合，容量制限は，各リンク $\kappa=1,\cdots,l$ に対して，

$$\sum_{i=1}^{q} x_\kappa{}^i \leqq c_\kappa \tag{20}$$

という形の線形不等式制約を課すことになるが，(16)，(20)（それに $x_\kappa{}^i \geqq 0$）により定められる凸多面体の頂点とその方向微分を計算し，さらに方向微分が最小であるような頂点を求めることは禁止的に複雑である．（制約式(20)のために各会社ごとに分離した問題にならない．）そこで，このような余分な制約に対しては，適当な罰金関数 $\bar{\varphi}(p,y)$ を用いて，目的関数を $z(\boldsymbol{x})$ から

$$\bar{z}(\boldsymbol{x})=z(\boldsymbol{x})+\sum_{\kappa=1}^{l}\bar{\varphi}(p,\ \sum_{i=1}^{q} x_\kappa{}^i-c_\kappa) \tag{21}$$

に変えて，$\bar{z}(\boldsymbol{x})$ を最小にする問題を解けば，制約(20)を近似的に満足する解が得られる．

$\varphi(p,y)$ としては，$\varphi(p,y)$ の値も $\partial\bar{\varphi}(p,y)/\partial y$ の値も，y が負のときには小さく，y が正になると急激に増大するようなものを用いなければならない．（p は，罰金の激しさを示すパラメタとする．）理論的には

$$\left.\begin{aligned}\bar{\varphi}(p,y)&=\frac{1}{p}e^{py},\\ \frac{\partial\bar{\varphi}(p,y)}{\partial y}&=e^{py}\end{aligned}\right\} \tag{22}$$

などがわかりやすいが，指数関数は計算に時間を喰うので，実用上は

$$\left.\begin{aligned}\bar{\varphi}(p,y)&=\begin{cases}(py+1)^2/2p & (y\geqq-\dfrac{1}{p}),\\ 0 & (y\leqq-\dfrac{1}{p}),\end{cases}\\ \frac{\partial\bar{\varphi}(p,y)}{\partial y}&=\max(0,py+1)\end{aligned}\right\} \tag{23}$$

という関数が便利である．

罰金関数を使うときには，p が大きければ大きいほどもとの問題への近似度が高くなることはよく知られているが，ＩＡ法と組み合わせると（他の方法と罰金法を組み合わせるときも同様なことが起こるが），p が大きいと§3の注意に述べたような問題が生じる．すなわち，p が大きくなると $\partial^2\varphi/\partial y^2$ が大きくなり，それは $\bar{z}(\boldsymbol{x})$ の導関数の Lipschitz 定数（あるいは，大ざっぱにいえば，$\bar{z}(\boldsymbol{x})$ の第2次導関数）を大きくし，ε を小さくとらなければならなくなる．つまり，罰金法の近似度を上げようとすると ε を小さくとって計算に手間をかけなければならなくなり，反対に，収束速度を上げるために ε を大きくとろうとすると p をほどほどに止めて近似度は悪くて我慢しなければならない，ということになるわけである．

道路容量の絶対的な制約条件をそれよりやや緩かな罰金関数でおきかえるという物理的な発想にしたがえば，「もしある罰金関数を用いて得られた解が制約条件を満足していればそれでよく，もし満たしていなければ—罰金関数を嶮しくするのも一つの手ではあろうがもう一つの手として考えられるのは—容量が仮にもっと小さかったとして再び解いてみたらどうであろうか」という着想は比較的素直なものであろう（“容量修正法”！）．より一般的には，制約式

$$g_k(\boldsymbol{x})\leqq 0 \qquad (k=1,2,\cdots) \tag{24}$$

を近似するために目的関数を，

$$\bar{z}(\boldsymbol{x})=z(\boldsymbol{x})+\sum_k\bar{\varphi}(p,g_k(\boldsymbol{x})) \tag{25}$$

として得られた解を $\bar{\boldsymbol{x}}(p)$ として，つぎに

$$\bar{\bar{z}}(\boldsymbol{x})=z(\boldsymbol{x})+\sum_k\bar{\varphi}(p,g_k(\boldsymbol{x})+g_k(\bar{\boldsymbol{x}}(p))) \tag{26}$$

を目的関数として得られた解を $\bar{\bar{\boldsymbol{x}}}(p)$ とする，……，というようにしたらどうかということである（**図5**）．実際，このような方法によれば，

① $\varphi(p,\cdot)$ が十分に嶮しければ $\bar{\bar{\boldsymbol{x}}}(p)$ は $\bar{\boldsymbol{x}}(p)$ よりもとの問題の最適解のさらによい近似になっていること，

② $z(\boldsymbol{x}),g_k(\boldsymbol{x})$ が線形である場合（すなわち，もとの問題が線形計画問題である場合）には，

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．

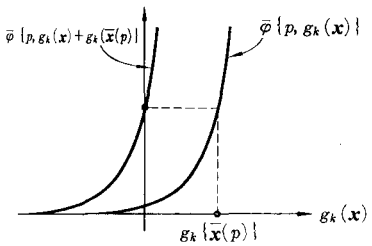


図 5 罰金関数の壁の移動(容量修正法)

$\bar{\varphi}(p, \cdot)$が十分嶮しければ$\bar{\bar{x}}(p)$はもとの問題の最適解と一致すること,

等が比較的容易に示される．このことはわが国においては［8］,［9］などに発表されただけでその後そのままにされていたが，それとは独立に同じ頃から世界的には“multiplier method”として一般的な形で研究が進められ，現在では非線形計画問題の数値解法としてもっとも有望なものの一つとして認められつつある．なお，multiplier method の一般論からは，①，②のほかにさらに

③$\bar{x}(p)$, $\bar{\bar{x}}(p)$, …の系列は($\bar{\varphi}(p, \cdot)$の嶮しさには無関係に）もとの問題の最適解に収束すること

も知られている．(multiplier method については［10］およびそこにあげられている文献表を参照されたい．)

おわりに

ＩＡ法は，そもそも物理的なシミュレーションという発想からはじめられたものではあるが，数理計画論の立場から見てもいろいろ興味ある点を含んだ方法であり，また，実用的にも，もっともっと多くの分野で手軽に使われてよい方法ではないかと思う．ＩＡ法の今後の発展に本稿がなんらかの寄与ができれば幸いである．

参考文献

［1］奥平耕造：ＩＡ法について，その発展過程と応用．オペレーションズ・リサーチ，22—12，688～694，1977.

［2］森口繁一：ＩＡ法における振動と収束．オペレーションズ・リサーチ，22—12，702～711，1977.

［3］J. G. Wardrop : Some Theoretical Aspects of Road Traffic. Research Institute of Civil Engineering, Road Paper No. 36, London, 1952.

［4］伊理正夫他：ネットワーク構造を有するオペレーションズ・リサーチ問題の電子計算機処理に関する基礎研究．日本オペレーションズ・リサーチ学会報文シリーズT-73-1，1973.

［5］西野吉次他：新手法による高速道路交通量の推計．日本オペレーションズ・リサーチ学会報文シリーズT-73-2，1973.

［6］M. Bruynooghe, A. Gibert et M. Sakarovitch; Une méthod d'affectation du trafic. W. Leutzbach und P. Baron (ed.), *Beiträge zur Theorie des Verkehrsflusses* (Referate anlässlich des IV internationalen Symposiums über die Theorie des Verkehrsflusses in Karlsruhe im Juni 1968), *Strassenbau und Strassenverkehrstechnik*, 86, 198～204, 1969.

［7］志水清孝：システム最適化理論．コロナ社(1976).

［8］森口繁一，伊理正夫，長谷　彰：多種流輸送問題に対する一つの逐次解法．1970年日本オペレーションズ・リサーチ学会秋季研究発表会論文集1-1-7，21～22，1970.

［9］森口繁一，伊理正夫，塚本幸生：多種流輸送問題の解法における容量修正法．1970年日本オペレーションズ・リサーチ学会秋季研究発表会論文集1-1-8，23～24，1970.

［10］藤重　悟，伊理正夫：SUMT から乗数法へ．電子通信学会誌，59—9，972～974，1976.

いり・まさを
1933年生
1955年東京大学工学部応用物理科卒
現在東京大学工学部計数工学科教授
専攻　数理工学

© 日本オペレーションズ・リサーチ学会．無断複写・複製・転載を禁ず．