文　献　抄　録

Appelgreen, Leif H., "A Column Generation Algorithm for a Ship Scheduling Problem, *Transportation Science*, **3**, 1 (1967), B-267—B-299

〔スケジューリング／線型計画／応用的〕

この論文は多数の船舶を世界中に配船しているある海運会社の配船計画作製のアルゴリズムについて研究したものである.

記号　i：船舶番号

k, j：積荷番号

n, m：積出し日

モデルは node (k, n) から構成されるネットワーク上でのフロー問題として考えられ，その流量は $x(i, j, m, k, n)$ で表わされる. $x(i, j, m, k, n)=1$ ならば，船 i が前の航海 (j, m) の後，次の航海 (k, n) についたことを意味するものとする. このモデルは次の線型計画によって記述できる.

$$\sum_i \sum_j \sum_m \sum_k \sum_n v(i, j, m, k, n)\, x(i, j, m, k, n) \rightarrow \text{最大化}$$

流れの制約：

$$\sum_k \sum_n x(i, j, m, k, n)=\sum_k \sum_n x(i, k, n, j, m),\ \forall i, j, m.$$

船舶についての制約：

$$\sum_k \sum_n x(i, 0, 0, k, n)=1, \qquad \forall i.$$

積荷についての制約：

$$\sum_i \sum_j \sum_m \sum_n x(i, j, m, k, n)\begin{cases}\leqq 1, \\ =1,\end{cases} \qquad \forall k.$$

(不等号は随意積荷の場合，等号は契約済積荷の場合)

非負条件：$x(i, j, m, k, n) \geqq 0,\ \forall i, j, m, k, n.$

ここで目的関数の係数 $v(i, j, m, k, n)$ は，

1. 随意積荷の場合の収益
2. 船の遊休時間にかかる premium
3. 船の航海・燃料費
4. 期末における船の位置による cost

などによって定められる.

以上の LP を解くのに，Dantzig-Wolfe の decomposition 法を利用する. すなわち，親問題としては積荷に関する制約を，子問題としては各船舶番号 i ごとに，流れ，船舶，非負についての制約をまとめて用いる. 子問題での目的関数は，

$$\sum_j \sum_m \sum_k \sum_n [v(i, j, m, k, n)-p(k)] \cdot x(i, j, m, k, n)-q(i)$$

となる. ここで，

$p(k)$：積荷番号 k に関する相対変数

$q(i)$：凸制約条件 $\left(\sum_{v=1}^{ki} \lambda_i^k=1\right)$ に関する相対変数

で親問題の解から与えられたものである. (k, n) に始点，終点を加えた network において，積出し日の順に node 番号 $s=0, 1, \cdots, S$ をつけ，$V(S)=0$ から始めて，各 node ごとに

$$V(s)=\max_{s'>s} \{V(s')+v[i, k(s), n(s), k(s'), n(s')] -p[k(s')]\}$$

を計算する. もし $V(0)-q(i)>0$ ならこの時の解が親問題に送られる. すべての船 i についての積荷順序から，親問題の LP が作られ，その時点での $p(k), q(i)$ を子問題に与える. 以上の step を繰返せばもとの問題の最適解に到達することができる.

船40隻，積荷50個の問題について，IMB 7090 による平均 2.5 分の計算により最適解が得られている.

このアルゴリズムでは解が整数解となる保証はないが，実際問題では全体の 1 ～ 2 % しか分数解は現われなかった. 整数解を得るための条件，整数型 LP として取扱う問題も検討されている. (山本正明)

Charnes, A. and W. W. Cooper, "Structural Sensitivity Analysis in Linear Programming and an Exact Product Form Left Inverse", *Naval Research Log. Quart.*, **15**, 4 (1968), 517—522

〔線型計画／係数の変る場合／理論的〕

行列 B の要素が一定数 σD (σ：スカラー，D：行列) だけ変化した行列 $B+\sigma D$ の逆行列を元の逆行列 B^{-1} と D, σ を用いて表現するということはよくやられているが，この論文では，(1)このことをいわゆる "一般化された逆行列" (ここでは left inverse だけについて論じられているが，同様のことは right inverse についてもなされる) について行っているもので，特に D に特殊な仮定を置くことによってその逆行列は大変簡単で応用上便利な形に

なることが示される．また(2)上の結果をLPに応用すれば，丁度基底行列の要素を変えたときの感度分析の方法になるわけで，そのときの各要素の変化の仕方が上の結果を用いて一般的に示されている．

いま，$B: m \times n$ の行列，$B^{\#}: B$ の左側逆行列 (left inverse) とし，D をある実数 p に対して $B^{\#}DB^{\#}D = pB^{\#}D$ が成り立つような $m \times n$ 行列とするとき，$\sigma \neq -p^{-1}$ に対して，

$$(B+\sigma D)^{\#} = B^{\#}(I+\tau DB^{\#}) = (I+\tau B^{\#}D)B^{\#},$$

ただし，$\tau = -\sigma(1+p\sigma)^{-1}$ が成り立つ．証明は D の性質（仮定）を用いて直ちに $(B+\sigma D)^{\#}(B+\sigma D) = I$ が得られる．また上のような性質をみたす D としては，ある1列だけを除いて他は全て0要素の行列や1つの要素だけが0でない行列などを挙げることができる．

B を LP の基底行列とすれば，$B+\sigma D$ は基底行列の変更になり，またその特殊なケースとして $\sigma = 1$, D を新しく基底に入る変数の係数ベクトルと追出される変数の係数ベクトルとの差を列ベクトル（ある1列）にもつ行列とすれば，$(B+\sigma D)^{\#}$ の計算は，丁度改訂シンプレックス法の1ステップの計算(up-dating) に対応することになる．

P_j を LP の第 j 変数の係数ベクトルとし，$Y_j = B^{\#}P_j$, $\pi^t = C_B^t B^{\#}$（C_B は基底変数の目的関数ベクトル）とすれば，基底行列の要素の変化に伴う各要素の変化は次のようになる．

$$\hat{Y}_j = (I+\tau B^{\#}D)Y_j$$
$$\hat{\pi}^t = \pi^t(I+\tau DB^{\#})$$
$$\hat{z}_j = z_j + \tau\pi^t DY_j$$

したがってこれらのことから基底行列の要素の変化するときの optimality range や feasibility range が求められる．（青沼竜雄）

Zionts, S., "The Criss-Cross Method for Solving Linear Programming Problems", *Management Science*, **15**, 7 (1969), 426—445.

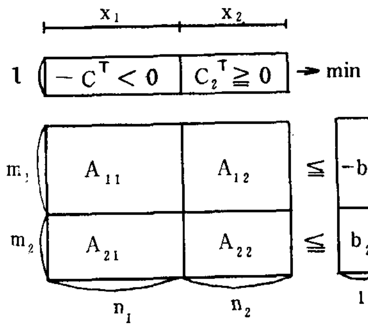
〔線型計画／アルゴリズム／理論的〕

LP の主シンプレックス法，双対シンプレックス法は，それぞれ，主許容な初期解，双対許容な初期解から始める．許容解が始めからはないときには人為変数を用いたりする．ここでは主許容解も双対許容解もないときに，人為変数を使うこともなく主シンプレックス法と双対法を交互に用いるアルゴリズムを提案している．

主許容でも双対許容でもないところに，主および双対法を用いるときには最適解に収束する証明がなかったので，この方法が考えられつつも確信をもって使われることはなかった．本論文には収束の証明がされている．

アルゴリズム 制約式の向きも，定数項の方が等しいか大きいという形に，反対向きの不等式には (−1) を掛けたりして，揃えておく．（等号条件式は計算中に，その式中の変数を基底に入れ不等式に直す工夫をする．）人為変数を用いないので，計算のタブローも単位行列を付けない形である．タブローの列は非基底変数，行が基底変数に対応する．各消去計算後に，タブローを，判定基準と定数項の符号によって行，列を入れ換えて整理して図1のようにしてみる．

図 1



ここで x_1, x_2 は非基底変数．書いてはない $m = m_1 + m_2$ の各行に対応するスラック変数が基底変数である．そして，上側の m_1 行が主非許容，左の n_1 列が双対非許容になっている．

下側の主許容な m_2 行の中から軸要素をみつけて消去演算をする主シンプレックス法の計算を主計算primal iteration, 双対許容な n_2 列の中から軸要素を選んで行う双対シンプレックス法の計算を双対計算 dual iteration と呼ぶ．

Criss-cross 法は，主計算と双対計算を交互に繰返してゆくシンプレックス法である．

この過程で，主（あるいは双対）変数が無限大の値で基底に入ることになったら，次の繰返しも主（あるいは双対）計算を行う．この計算ができないときには双対（主）計算にする．いずれもできないときには，この問題に有界な最適解が存在しないことが示される．

また主（あるいは双対）許容解に達したときには，それ以後は主（双対）計算のみを繰返せば最適解を得る．

もし多数回（たとえば $2m+2n$）の主，双対計算

の交互繰返しでも，いずれの許容解にもならないときには，制約式 $\sum x_j \leqq M$（十分に大きい数）を追加した上で，双対許容になるまで主計算のみを行う．そのあとで双対計算のみを繰返し，(1)最適解を得たときに，追加制約式が（1•1）厳密な不等号で成立っていれば元の問題に許容最適解があるが，（1•2）等号で成立っていれば有限許容解はない．また(2)双対問題の解が有界でなくなったときには，最適許容解は存在しない．実はこれまでの実験例では，追加制約式を用いるに至ってはいないらしいが，この式は収束の証明には必要なものである．

アルゴリズムは，細かい手順も実際には必要で，10ページにわたるフロー•チャートで詳述してある．

特徴 1．初期（主／双対）許容解がなくてもいい．

2．人為変数を使わない．

3．積型式のアルゴリズムを作れば使える．

4．普通のシンプレックス法より，計算時間，繰返し数共に節約できる実績がある．

さらに，積型式，特に大きい問題での実験が必要であるし，追加制約式のいらない収束の証明や，退化した場合の取扱いについて調べる必要があると結んでいる．（真鍋竜太郎）

Pierce, J. F., "Application of Combinatorial Programming to a Class of All-Zero-One Integer Programming Problems," *Management Science*, 15, 3 (1968), 191—209

〔整数値計画／組合せ分析／理論的〕

本論文は組合せ計画法における手順を（0--1）変数整数計画問題の解法に適用したものである．これはまず最初にある実行可能解を探し，つぎに逐次良好な実行可能解を探していき，最終的に最適解が得られるもので同様な手順をくり返すものである．このくり返し過程において問題の変数を bit で表わすことにより，2進計算機における論理和および論理積の処理により，数百位の変数を含むような問題も数秒で解けることを報告している．

ここで考えている問題はつぎのとおり．

Min. $Z=\sum_{j=1}^{j=n} c_j x_j$

Sub. $\sum_{j=1}^{j=n} x_j A_j = R$

$x_j=0$ or 1, $j=1,2,\cdots,n$

あるいは，

Min. $Z=\sum_{j=1}^{j=n} c_j x_j$

Sub. $\sum_{j=n}^{j=n} x_j A_j \geqq R$

$x_j=0$ or 1, $j=1,2,\cdots,n.$

ただし，A_j は要素が0か1からなる $m\times 1$ 行ベクトル $(a_{1j}, a_{2j},\cdots,a_{nj})$ であり，R は全要素が1であるような $m\times 1$ 行ベクトル $(r_1, r_2,\cdots,r_m)$ である．さらに全ての c_j は非負とする．

この種の問題の解法としては少くとも implicit には全ての可能な解について調べる必要があるが，その check の過程において dominance あるいは bounding，さらに feasibility の点で考慮しなくて済む組み合せについては省略してより効率的に最適解を求めるものである．したがって，本論文での解法 algorithm も，いわゆる "branch and bound 法" であり，"implicit enumeration 法" であり，また "tree search algorithm" である．この意味で，本質的には Egon Balas の Additive Algorithm と同じものであるが，ただ Additive Algorithm をより特殊な問題に適用して効率的に解を求めようとするものである．解 $X=(x_1,x_2,\cdots,x_n)$ を lexicography に逐次 chech してゆき，いま s-1 stage で，$\bar{x}_1,\cdots,\bar{x}_{s-1}$ が定められたとすると残りは，

Min. $\sum_{j=s}^{j=n} c_j x_j$

Sub. $\sum_{j=s}^{j=n} x_j A_j = R-\sum_{j=1}^{j=s-1} \bar{x}_j A_j$

$x_j=0$ or 1, $j=s, s+1,\cdots,n$

なる問題につき考える．ここで feasibility につき調べる手順として，

（i） $x_1=F(R_1), x_2=F(R_2),\cdots,x_n=F(R_n)$ とする．もし $R=\sum_{j=1}^{j=n} x_j A_j$ ならば X を best feasible solution として $X°$ とし，$Z°=\sum_{j=1}^{j=n} c_j x_j°$ として（iii）に進む．

（ii） もし，$x_j=1$ となるような j が存在すれば，そのような j の最大値を S として（iii）に進む．そうでないときは stop．

（iii） $x_1=x_1, x_2=x_2,\cdots, x_{s-1}=x_{s-1}$, $x_s=0$, $x_{s+1}=\mathrm{F}(R_{s+1})\cdots, x_n=F(R_n)$ とする．

$R=\sum_{j=1}^{j=n} x_j A_j$, $\sum_{j=1}^{j=n} c_j x_j < Z°$ であれば X を best feasible solution として store しておき（ii）に進む．

ただし，$R_s=R-\sum_{j=1}^{j=s-1} x_j A_j$

$$F(R_s)=\begin{cases}1 & R_s\otimes A_s=A_s\\ 0 & \text{その他}\end{cases}$$

ここで記号⊗は論理積の演算を表わすものとする．

さらに bounding test（これは Balas の algorithm における ceiling test に相当している）をおこない，必要な elimination を考えている．また dominance の概念を導入することにより elimination を考えているが，この過程は Fred Grover の surrogate constraint の考え方と同様である．

以上のような考え方で実際問題を解いた計算例があるが，例えば $m=8$, $n=92$ であるようなトラック発送の問題は 0.5 秒程度，$m=12$, $n=298$ で35秒程度の時間で IBM 7094 で解いている．（成久洋之）

Nemhauser, G.L., and D.A. Pierce, "Computational Results for a Stopping Rule Problem on Averages," *Naval Research Logistics Quartery,* 15, 4 (1968), 567—578.

〔確率論／ダイナミック・プログラミング／応用〕

Optimal stopping rule に関する論文は，1963年頃より多く見られるが，stopping rule の存在性をマルチンゲールなどを用いて考察する純理論的なものと，ダイナミック・プログラミングを用いた計算手法を与えるものとに大別できるが，この論文は後者に属するものである．

互に独立で，同じ分布に従う確率変数列を，X_1, X_2, $\cdots$ とし，$P_r(X_1=1)=P_r(X_1=-1)=1/2$, $S_n=X_1+X_2+\cdots+X_n$ とおく．この確率変数を逐次観測し，n 回の観測の後，stop して利得 S_n/n を受けとるか，または，次の確率変数を観測するかを決定する．この決定の criterion として $E(S_n/n)$ を最大にするような有限 stopping rule が存在することは，Y.S. Chow と H. Robbins によって最初に証明され，A. Dvoretzky が一般化している．上述の問題を P とすると，P が解をもつための必要十分条件は，問題 $\{P_N;N=1,2,\cdots\}$ なる系列が収束することであり，その極限値がPに対する解になる．P の optimal rule の構造は，

$$S_n \geqq k_n \rightarrow \text{stop}$$

$$S_n < k_n \rightarrow \text{continue}$$

なるような正整数 $k_n \leqq k_{n+1}$ で表わされ，P_N の optimal rule は

$$S_n \geqq k_n^N \rightarrow \text{stop}$$

$$S_n < k_n^N \rightarrow \text{continue}$$

なるような整数 k_n^N で特徴づけられる．従って，P の近似解は，適当に大きい（有限）P_N を解くことによって得られ，しかも

$$k_n^N \leqq k_n^{N+1} \leqq \cdots \leqq k_n$$

となっている．

すなわち，P_N の解は k_n の下限を与えている．この論文では，k_n の上限を考え，下限と上限が一致した場合にその値を計算することによって optimal policy を見つける方法が用いられている．

論文の前半は Y.S. Chow と H. Robbins のもの* とほとんど内容は重複しており，新鮮味はないが，計算にダイナミック・プログラミングを用いているのと，k_n に対する上限が与えられており，最後の数ページでは数値計算の結果が表とグラフを用いて詳細に述べられている点で，応用に役立つであろう．

* Y.S. Chow and H. Robbins; On Optimal Stopping Rules For S_n/n, Illinois Journal of Math. Vol. 9, 1965 （田畑吉雄）

Zangwill, W.I., "A Backlogging Model and a Multi-echelon Model of a Dynamic Economic Lot Size Production System—A Network Approach," *Management Science,* 15, 9 (1969), 506–527.

〔生産／ネットワーク／理論的〕

経済ロット・サイズ問題に関して，従来二つのアプローチがなされている．一つは，すでに1915年頃から研究されてきた古典的，または Wilson の経済ロット・サイズ公式として知られているものであり，そこでは，期間を通じて需要速度は一定という条件の下に，微分法により，最適な生産スケジュールを決定しようという方法がとられている．一方，他の一つは，経済ロット・サイズ問題に対するダイナミック・アプローチといわれるもので，そこでは期間をいくつかの離散的なピリオッドに分割し，ピリオッド毎の需要速度は必ずしも一定ではないという条件のもとに，数理計画法により，最適生産スケジュールを求めていこうとするものである．本論文では，後者の立場から，二つのモデルについて検討がなされている．すなわち，第一に，Wagner, Whitine のモデルをバックログが許される場合にまで拡張し，こうした場合に対する新しいダイナミック・プログラミング・アルゴリズムを提出している．そして，第二のモデルとして，工程が多段階の場合を考え，この場合にも，第一のモデルと同様に，コンケイブ・コスト・ネットワーク分析を適用して，全ての段階での最適な生産スケジュールを求めるための効果的なダイナミック・プログラミング・アルゴリズム

を提出している．以下に，本論文で提出された第一のモデルに対するダイナミック・プログラミング・アルゴリズムの概略を紹介する．

$r_i(r_i \geqq 0)$ をピリオッド i における需要量，$x_i(x_i \geqq 0)$ をピリオッド i における生産量，I_i をピリオッド i 期末の在庫量とし，x_i 単位生産するための費用を $P_i(x_i)$，ピリオッド i における在庫量 I_i を保管するための費用を $H_i(I_i)$ とすれば，これらの費用は $[0, +\infty)$ の範囲でコンケイブ（凹）になると仮定される．したがって最適な生産スケジュールを決定するには，

$$(1) \quad \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} r_i, \qquad (2) \quad x_i + I_{i-1} - I_i = r_i$$

$$(3) \quad I_0 = I_n = 0,\ x_i \geqq 0,\ I_i \geqq 0$$

なる条件で，

$$(4) \quad \sum_{i=1}^{n} P_i(x_i) + \sum_{i=1}^{n} H_i(I_i)$$

を最小化するような x_i を求めればよい．

ここで著者は，シングル・ソース・ネット・ワークの概念を適用している．すなわち，目的関数がコンケイブで，シングル・ソースをもつネット・ワークでは，それぞれのフローにおける各ノードは少なくとも一つの正なるインプットをもつ，最適生産スケジュールはこうしたフローの中での最適なフローにほかならない．ところで，バック・ロッグが許される場合には，ピリオッド i における在庫量は負の値をとりうる．したがって，ピリオッド i における品切れ分を I_i^-，ストックされている在庫量を I_i^+ として表わすことにし，それぞれの費用を $H_i^-(I_i^-)$，$H_i^+(I_i^+)$ とすれば，前述の制約条件式及び目的関数はそれぞれ次のように書きかえられる．

$$(1') \quad \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} r_i$$

$$(2') \quad x_i + I_{i-1}^+ - I_{i-1}^- - I_i^+ + I_i^- = r_i$$

$$(3') \quad I_0^+ = I_0^- = I_n^+ = I_n^- = 0,\ x_i \geqq 0,\ I_i^+ \geqq 0,\ I_i^- \geqq 0$$

$$(4') \quad \text{Minimize}\left\{\sum_{i=1}^{n}(P_i(x_i) + H_i^+(I_i^-) + H_i^-(I_i^-))\right\}$$

ここで，任意のピリオッド i における在庫 I_i が $I_i=0$ なる時をリジェネレーション・ポイントと定義すれば，最適スケジュールは，二つの生産が行なわれるピリオッドの間に一つのリジェネレーション・ポイントをもつ，αを一つのリジェネレーション・ポイントとし，β を次ぎに生産が行なわれるピリオッド，γ を α の次のリジェネレーション・ポイントとすれば，ピリオッド $\alpha+1$ から n までの最適費用 a_α，ピリオッド $\beta-1$ に需要が ε ピリオッド・バックログされているものとしてピリオッド β から n までの最適費用 $b_\beta(\varepsilon)$ はそれぞれ次のようになる．

$$(5) \quad a_\alpha = \underset{\mu<\beta \leqq n}{\text{Min}}\left\{\sum_{k=\mu+1}^{\beta-1} H_k^-\left(\sum_{i=\mu+1}^{k} r_i\right) + b_\beta(\beta-\alpha-1)\right\}$$

$$(6) \quad b_\beta(\varepsilon) = \underset{\beta \leqq \gamma \leqq n}{\text{Min}}\left\{P_\beta\left(\sum_{k=\beta-\varepsilon}^{\gamma} r_k\right) + \sum_{k=\beta}^{\gamma-1} H_k^+\left(\sum_{i=k+1}^{\gamma} r_i\right) + a_\beta\right\}$$

ここで，生産費の中での固定部分を考慮すると上式は，次ぎのように書きかえられる．

$$(5') \quad a_\alpha = \underset{\mu<\beta \leqq n}{\text{Min}}\left\{\sum_{k=\mu+1}^{\beta-1} H_k^-\left(\sum_{i=\mu+1}^{k} r_i\right) + b'_\beta\right\}$$

$$(6') \quad b_\beta' = \sigma_\beta + \underset{\beta \leqq \gamma \leqq n}{\text{Min}}\left\{\sum_{k=\beta}^{\gamma-1} H_k^+\left(\sum_{i=k+1}^{\gamma} r_i\right) + a_\gamma\right\}$$

最適生産スケジュールは，(5′)，(6′) 式を用いることにより，$a_n=0, b'_{n+1}=0$，としてピリオッド n からバック・ワードに決定される．なお，(5′)，(6′) 式は計画期間定理を用いることにより計算手続が簡略化される．　　(田中芳彦)

Newhauser, G.L., and Z. Ullmann, "Discrete Dynamic Programming and Capital Allocation," *Management Science*, 15, 9 (1969), 494—505

〔投資／数理計画／応用的〕

T期間にわたって，利用可能な資金の制限のもとで，採用されるプロジェクトの正味現在値の和が最大となるように投資プロジェクトを選択するという問題を考える．ここでプロジェクトは分割不可能なものと考え，この問題を dynamic programming を使って解くことを試みる．問題の定式化は Weingartner [2, 3] と同じであるが，計算の仕方がいくらか異なっている．すなわち，利用可能な資金量が，$I=(i_1, \cdots, i_T)$ であるとき，プロジェクト $P_1, \cdots, P_k$ からの最大利益を $f_k(I)$ とすると，$f_k(I)$ はステップ関数で，しかも I の単調増加関数となるという事実を使って，すべての I について $f_k(I)$ を計算，記憶する代りに，$f_k(I)$ の値が変化する I についてだけ $f_k(I)$ を求めることを考える．$J_{kn}=(j_{1kn}, \cdots, j_{Tkn})$，$n=1, \cdots, Z_k$ （Z_k は $f_k(I)$ の値が変化する点の数）で $f_k(I)$ の値が変ったとし，プロジェクト P_{k+1} の毎期の資金の必要量が $A_{k+1}=(a_{1k+1}, \cdots, a_{Tk+1})$，その正味現在値が b_{k+1} であると，(f_k

$(J_{kn})+b_{k+1}$, $J_{kn}+A_{k+1})$ と $(f_k(J_{km}), J_{km})$ $(m, n=1, \cdots, z_k)$ を比較して，より大きな資金量でより少ない利益しかもたらさないような組み合わせは除いて，残った $(f_k(J_{kn})+b_{k+1}, J_{kn}+A_{k+1})$ と $(f_k(J_{km}), J_{km})$ から $f_{k+1}(I)$ を求める．この計算法は I の次元が小さい（計画期間が短かい）ときには有効であるが，I の次元が大きくなると計算の複雑さは急速に増加する．著者は5期間で50プロジェクトを取り扱うプログラムを作成しているが，期間が長くなったときには Geoffrion [1] らの implicit enumeration が dynamic programming より有効となることを指摘している．

なお，かれらは，各プロジェクト $P_k(k=1, \cdots, N)$ が L_k レベルのうちどれかのレベルで採用することができ，レベル m で採用すれば正味現在価値が b_{km}，資金必要量が $A_{km}=(a_{1km}, \cdots, a_{Tkm})$ となる多数レベル・プロジェクトの問題，投資プロジェクトからの収益の再投資を認める問題，借入れおよび貸付を認める問題，ある期で利用可能な資金を全部使用しなかったとき，それを次の期に繰り越して利用することを認める問題，プロジェクト相互が独立でなく，資金の必要量，正味現在価値が相互依存的である場合の問題について，いずれも上の計算法が適用できることを示している．プロジェクトが相互依存的である場合については，プロジェクト P_k とプロジェクト P_m を採用したとき，資金の必要量が $V_{km}+A_k+A_m$ 正味現在価値が $d_{km}+d_k+d_m$ という形で表わすことができる場合について考察している．最後にかれらが行った計算結果が紹介されている．この論文は Weingartner の線に沿って，その計算法の改良と，モデルの拡張が考えられているが，かれらも指摘するように計算法に関しては，期間が短かい場合には有効であるが，期間が長い場合には適当ではない．また，モデルの改良点は，比較的小さい点であって，正味現在価値を使用することの可否といった根本問題については検討されていない．

文　　献

1. Geoffrion, A.M., "An Improved Implicit Enumeration Approach for Interger Programming," *Operations Research*, **17**, 3 (May-June, 1969), 437—454.
2. Weingartner, H.M., "Capital Budgeting of Interrelated Projects: Survey and Synthesis," *Management Science*, **12**, 7, (March 1966), 485—516.
3. ——, and D.N. Ness, "Methods for the Solution of the Multi-Dimensional 0/1 Knapsack Problem," Operations Research, **15**, 1 (January-February, 1967), 83—103. (飯原慶雄)

Ignall, E., and A.F. Veinott, Jr., "Optimality of Myopic Inventory Policies for Several Substitute Products," *Management Science*, **15**, 5 (1969), 284—304

〔在庫／最適化／理論的〕

多種類の製品を含む在庫問題において，費用としては，購入費用が購入量に比例し，発注の際の固定費がかからないとした時，1期間のみを対象にした場合の最適政策が N 期間を対象に考えた場合の（定常な）最適政策になるための条件を求めている．前に Veinott が同種の問題を扱っているが，今度は製品の間に代替性（substitute property）を入れることによって，その条件をゆるめることを主な結果としている．

製品は n 種類，需要は m クラスとし，次の記号を用いる．

$D_t=(D_{t1}, \cdots, D_{tm})$：時刻 t の需要ベクトル，$D_1, D_2, \cdots$ は独立，同一同時分布（Φ）を持つ．

$x_t=(x_{t1}, \cdots, x_{tn})$：時刻 t の製品の発注前の在庫量

$y_t=(y_{t1}, \cdots, y_{tn})$：時刻 t の製品の発注後の在庫量

$$x_{t+1}=s(y_t, D_t), \quad t=1, 2, \cdots$$

s は与えられたボレル関数で，例えば $n=m$, backlog の場合には $s(y_t, D_t)=y_t-D_t$.

購入ベクトル z に対して購入費用は，

$$c(z)=\sum_{j=1}^{n} c_j \cdot z_j$$

発注後の在庫量 y_t で需要 D_t のときの t 期の在庫費用，品切れ費用は $g(y_t, D_t)$ とする．不足分は backlog し，納入おくれはない，とする．

$$G(y)=\int_D [cy+g(y, u)-\alpha c \cdot s(y, u)] d\Phi(u)$$

とおき，$f_N(x_1|\overline{Y})$ を初期在庫 x_1 で，政策 $\overline{Y}=(\overline{Y}_1, \overline{Y}_2, \cdots)$ を用いたときの N 期間の割引き総費用の期待値とすると，

$$f_N(x_1|\overline{Y})=\sum_{t=1}^{N} \alpha^{t-1} EG(y_t)$$

となり，このとき，$f_N(x_1|\overline{Y}^*)=\min_{\overline{Y}} f_N(x_1|\overline{Y})$ であ

れば，$\bar{Y}^*$ が $x_1 \epsilon X$ に対して N-optimal であると定義する．

仮定 I：$y(s(\bar{y}(x),u))=\bar{y}(s(w(x),u)) \quad x \epsilon X, u \epsilon D$

仮定 II：$v^\vee s(y,u)$ は，固定した $u \epsilon D$ に対して，$[v,\infty] \cap X \ni y$ に関して，非減少関数である．

ここで，$w(x)$ は初期在庫 x の時の発注後の在庫量 y の下界，v は x に関する $w(x)$ の下界，すべての x に対して $v \leqq w(x)$．このとき次の定理が成り立つ．

定理 仮定 I，II が成り立つならば，1-optimal な政策はすべての $x_1 \epsilon X$，すべての N に対して N-optimal となる．

つぎに，この仮定 I，II はどんな場合に成り立つかを考える．

まず，$\bar{y}(x)-x$ が $X \ni x$ に関して非増加関数のとき「$\bar{y}(\cdot)$ が代替性を持つ」と定義する．これは，ある一つの製品 i の初期在庫を増すと，他の製品 $j \neq i$ の発注後の在庫を減少させることで，製品 i の在庫が製品 j の在庫を代替したことになる．

仮定 III：$x \leqq x' \leqq \bar{y}(x)$ ならば，$\bar{y}(x)=\bar{y}(x'), x, x' \epsilon X$

仮定 IV：$v^\vee s(y,u)=v^\vee(y-r(u)) \quad y \epsilon X, u \epsilon D$

$r(\cdot)$ は D から R_n^+ へのボレル関数．

定理 仮定 III，IV が成り立ち，$\bar{y}(\cdot)$ が代替性を持つならば，仮定 I，II が $w(x)=v^\vee x, x \epsilon X$ でなりたち 1-optimal が，すべての N とすべての $x_1 \epsilon X$ に対して N-optimal となる．

つぎに，$\bar{y}(\cdot)$ が代替性を持つための $G(\cdot)$ に関する十分条件を求める．

substitute matrix の定義

H を $n \times n$ 行列とし，H^{ij} を H の i 列と j 列を交換した行列とする．H が対称で正定符号で，i 列，j 列の中のちょうど 1 つの要素を含むような H^{ij} の主小行列式が，すべての $i<j$ に対して非負である時，H は substitute matrix という．

$C^k(Y)$ は Y を含む開集合の上で k 回連続微分可能の関数の集り．$\nabla^2 G(y)=(D_{ij}G(y))$ は 2 次の偏導関数の Hessian 行列とする．

定理 $a, b \epsilon R_n^*$, $a \leqq b$, $Y=[a,b]$, $G \epsilon C^2([a,b])$, $\nabla^2 G(y)$ がすべての $y \epsilon [a,b]$ に対して，substitute matrix であるならば，$\bar{y}(\cdot)$ は代替性を持つ．

さらに，これ等の条件を，貯蔵量に制限のある場合や，multiechelon モデルに応用している．また，納入遅れが一定期間で backlog のある場合や，パラメタが期に依存する非定常の，あるモデル（定常モデルの translate である場合）についても述べている．（反町迪子）

Hardgrave, W. W., and W. H. MacWilliams, "Management Science Applications in the Bell System," *Management Science*, **15**, 8 (1969), B-387–396.

〔通信／最適化／応用的〕

本論文では，ベル・システムが施設，運用それに営業方針―顧客の需要構造の把握と提供できるサービスの検討，業界又は国家的な見地からの考慮を払ったサービスと料金の関係―の 3 分野について，経営における科学的諸手法を応用して行った検討結果を例をあげて述べている（ただし，方法そのものについては触れていない）．

施設への応用の分野では，交換局設置の問題を例にとり，最適配置，コスト／パーフォーマンス等に関して充分実用に供せる計算機のプログラムがすでに完成しておりこの種の開発はかなり進んでいる．

運用への応用の分野では，予め定めたサービス基準を維持して最小のコストとなるような運用方法を決定するのが問題となるが，この種の問題は通常昔どおりの解析では出来ない程複雑となるのでシミュレーション等を行なっているが，近似的な解しか得られない事がしばしばである．

また，人的な面の解析も試み，運用の一つのファクターとして組み込もうと努力している．

営業方針の決定の分野は，これが基礎となって事業計画の方向が定まるので最も重要なものではあるが，前二項にくらべて開発は遅れている．顧客の要求を満たすシステムの設計については，現在計算機のプログラムのアルゴリズムは開発の途上であってこれで求めたものが人手で作ったものより安いことは稀であるが，複雑なものなら人手よりはるかに短時間で解が求まる．

今後の問題として残されているものは，方法論的な困難性の打破―より良い解法の発見，膨大な規模の克服―であり，これにより，更によい解と，ますます重要性が増す営業方針の問題の解を得ることである．（牛塚清吾）