論文紹介

整数計画法

M10　分数ナップサック問題

H. Ishii, T. Ibaraki & H. Mine. 255-271.
Mathematical Programming **13**, 1977.

目的関数として分数式(ただし，分母分子ともに一次式)をもち，線形制約式が一つであるナップサック型問題が取り扱われている．

アルゴリズムは，非線形分数計画用に開発された，ディンケルバッハのアルゴリズムを改良している．通常の分数計画と同様に，目的関数$N(X)/D(X)$に対し，パラメータλを導入して$N(X)-\lambda D(X)$を最大化する，ナップサック問題を解く問題に変換する．λを$N(X)/D(X)$のつくる系列として，ナップサック問題を繰り返すわけである．

この際に，greedy 解がナップサック問題のよい近似解を与えることを利用して効率をあげている．また，制約式$\sum c_j X_j \leq d$のdの値によって，問題のサイズを事前に縮少するようなことも考察されている．

ディンケルバッハのアルゴリズムに対し，修正されたアルゴリズムが最悪の場合においても，劣らないということと，両アルゴリズムの繰返しの回数の上限が最後に求められている．　　　　　　　　　　(平林隆一)

M11　グラフのクリティカル・カットセットと集合パッキング多面体のファセット

E. Balas & E. Zemel. 15-19.
Mathematics of OR **2**, 1977.

この論文は集合パッキング問題とよばれる特殊な 0-1 整数計画問題をとり上げ，その実行可能領域に含まれる整数格子点の定める凸多面体のフェイス構造を研究している．ここではとくに Chvàtal が得たフェイスのための充分条件を拡張し，$\sum_{j\in S} x_j \leq k$のタイプの不等式がフェイスとなるための充分条件と必要条件を与えている．この種の研究は整数計画法(組合せ計画法)における流行の一つであり，同一号にも Wolsey による同系統の論文が見られることを付け加えておく．　　　　　(今野　浩)

確率統計応用

P5　二つの状態からなるセミ・マルコフ過程の仮説検定

A.K.Banerjee & G.K.Bhattacharyya. 340-356.
Sankhyā **38**, 4, 1976.

信頼性における故障・劣化モデル，待ち行列，在庫問題というような確率現象のモデルとして，セミ・マルコフ過程はよく使われていて，確率過程論的観点からの研究はよく行なわれている．しかしながら，セミ・マルコフ過程のモデルを実際に応用する場合には，データから推移確率行列，滞留時間の分布等を推定することが重要となるが，このような統計的観点からのセミ・マルコフ過程の研究はほとんどみられない．本論文は，セミ・マルコフ過程を統計的側面から研究したものである．

$F_{ij}(x)\,(i, j=1, 2)$を，状態iから状態jへの推移時間分布とする．また，$F_{ij}(x)=F_i(x)$として，その密度関数は$f_i(x)=c(\theta_i)\exp\{\theta_i w(x)\}h(x)$，$x\geqq 0$とあらわされるものとする．このとき，観測データ$\boldsymbol{R}=(S_0,\ X_1,\ S_1,\ X_2,\ \cdots,\ X_N,\ S_N)$($X\cdots$滞留時間，$S\cdots$状態)から，つぎの仮説検定

$$H_0 : \theta_1=\theta_2,\quad H_1 : \theta_1>\theta_2$$

を考える．

$f_i(x)$が指数型分布族の一つであることを利用して，

1. 状態 2 への推移がν(定数) 回観測されたときのデータ$\boldsymbol{R}$(Nが確率変数となる)
2) データの観測数Nを固定したときのデータ$\boldsymbol{R}$ (νが確率変数)

の二つの場合に，上記の仮説の一様最強力検定方式を導くとともに，その検出力について調べている．この結果を$f_i(x)$が指数分布の場合に応用するとF検定が導かれることを最後に例として示している．　　　(宮村鐵夫)

コンピュータとシミュレーション

C2　計算機ソフトウェアのデバッグおよびテストのための経験的停止規則

E.H.Forman & N.D. Singpurwalla. 750-757.
J. Amer. Stat. Assoc. **72**, 360, 1977.

信頼性理論の考え方を使って，プログラム中の誤りの数を推定し，デバッグおよびテストを終了するための判定基準を提案している．モデル：「誤りが発見される確率は，プログラムに含まれている誤りの数に比例し，誤りが発見される(CPU時間軸上での)時点の間隔は指数分布をする．発見された誤りは完全に取り除かれる」．誤りの数は最尤法によって推定する．尤度関数の形と漸近分布 (正規分布) の形とを比較して，両者が近ければ最尤推定値は妥当であると考え，その時残っている誤りの個数の推定値が 0 であればテストを終了する．3000～4000行の FORTRAN プログラムに対してこの方法を適用した結果が述べられている．　　　　(伏見正則)