論文紹介

スケジューリング

1　先行関係を含み線形または割引き費用を有する一機械スケジューリング問題について

K. D. Glazebrook & J. C. Gittins 161-173.
Operations Research **29**, 1, 1981

1台の機械を用いて，仕事の集合 $J=\{1,2,\cdots,K\}$ を処理する．仕事 i の処理時間は P_i で，i が時間 F_i で完了すればその費用は $C_i(F_i)$ である．集合 J 上には先行関係Rがあり，$(i,j)\in R$ ならば仕事 j を処理し始める前に仕事 i を完了しておかねばならない．(実行)可能な置換とは R に矛盾することなく J の仕事を処理する順序付けのことである．費用として $C_i(F_i)=-w_i a^{F_i}(w_i>0,\ 0\leq a<1)$ という割引き費用を考えた時，この費用の総和：$TC(\alpha)=\sum_{i=1}^{K}C_i[F_i(\alpha)]$ を最小にする可能な置換 α を見つける問題を考える．J の部分集合 U が初期集合と言われるのは，仕事 i とこれを先行する任意の j との間に $i\in U\to j\in U$ となっている時である．また J の部分集合 U に対して実数値関数 $\rho(U)$ を定義する．

さて，本論文ではネットワーク(J,R)に対して極小集合の中で$\rho(U)$を最小にする初期集合 U が存在すれば U の仕事を先に処理しそれから $J-U$ の仕事を処理する最適置換が存在することを前半で述べている．

論文の後半では処理時間 P_i が有限で正の確率変数となった場合を考えている．P_i の分布は既知で P_i と $P_j(i\neq j)$は独立と仮定する．(実行)可能な戦略とはRに矛盾することなく J の仕事を処理する任意の規則である（つまり中断も可）．ここでは費用の総和の期待値：$TC(\pi)=E\{\sum_{i=1}^{K}C_i[F_i(\pi)]\}$を最小にする可能な戦略 π を見つけるというよりはむしろ論文の前半で述べられた最適置換を見つける方法が用いられるためには，P_i に関してどういう条件があればよいかを示している．（一森哲男）

待ち行列

2　優先客の飛び越しサービスが可能な $M/M/1$ 待ち行列

E. Kofman & S. A. Lippman 174-188.
Operations Research **29**, 1, 1981

優先客と普通の客の2種類の客がある，$M/M/1$ 待ち行列を考察している．この2種類の客は，互いに独立なポアソン過程にしたがって到着し，その順序で1列に並ぶ．2種類の客は，待ち行列に並んでいる時にかかる保管費用によって区別される．扱者は，到着順にサービスをするか，もしくは，Rを払って，優先客を先にサービスする．扱者は，単位時間当りの期待費用を最少にするように客をサービスする．この時，（優先客を先にサービスするために飛び越される普通の客の数)×(1サービス期間当り，優先客1人当りの期待保管費用)が R 以上ならば，優先客を先にサービスするのが厳密に最適である．また，有限待ち行列の場合には，Rが増加すれば，優先客を先にサービスするのが最適である状態の集合は減少する．等が示されている．（行方常幸）

スケジューリング

3　同時に複数台のプロセッサを必要とするタスクシステム

E. L. Lloyd 189-201.
Operations Research **29**, 1, 1981

この論文は，通常のタスクシステムモデルの1つの拡張をとり扱っている．この拡張は，各タスクが，それらの実行中の各ステップにおいて，複数台のプロセッサを必要とすることである．同時性(concurrency)をもつタスクシステム $\mathscr{S}=(\mathscr{T},<,m,\mathscr{C})$は，タスクの集合 $\mathscr{T}=\{T_1,T_2,\cdots,T_n\}$，タスク間の先行関係，$m$ 台の等価なプロセッサ，を含んでいる．各タスク T_i の実行時間は正の整数 τ_i で，同時性の度合は $q_i\in\mathscr{C}$(ただし，$\mathscr{C}\subseteq\{1,2,\cdots,m\}$) である．各タスク T_i は，τ_i ステップのあいだ実行され，その間に q_i 台のプロセッサを必要とする．

この論文では，すべてのタスクの実行時間が等しい場合（concurrent UET タスクシステム）に，最大完了時刻を最小にするスケジューリング問題に対して3つの結果を与えている．

まず最初に，プロセッサが3台だけであり，各タスクの同時性が1または2の時ですら，concurrent UET タスクシステムスケジューリングは *NP*-complete であることが示される．

2番目には，同時性の最大値が r である時に，任意のリストスケジュールの値と，最適スケジュールの値の比が $(2m-r)/(m-r+1)$で上から抑えられることを示し，その比が $\lfloor(2m-r)/(m-r+1)\rfloor$ の時には，その比を実現する例を与えている．

最後に，プロセッサが2台の時に，最適スケジュールを構成するアルゴリズムを与えている．（益田照雄）

© 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．