機械スケジューリング問題の近似解法

木瀬　洋

1. はじめに

“複数個の仕事を1あるいはそれ以上の台数の機械で処理するとき，各仕事についてそれを処理する機械および各機械についてそこで処理される仕事の順序，すなわちスケジュールを決定する問題”は機械スケジューリング問題(machine scheduling problem) といわれている．多くの場合，種々の制約条件を満たして与えられた評価関数の最適値を達成するスケジュールを求める最適化問題が考えられる．

上記でいう機械は単なる機械のみならず人，施設等，仕事を処理する手段を意味するのであらゆる分野においてこの問題をモデルとするスケジューリング問題が存在する．しかしながらより多くの機械スケジューリング問題が種々の生産工場や計算機システムでおきる．特に多品種少量生産が多くなった昨今ではスケジューリングは生産計画の重要な地位を占めているものと考えられる．

ところで最近の NP 完全理論によって機械スケジューリング問題を厳密に解くことは，ほとんどの場合膨大な計算を要するので実際上不可能なことが明らかにされている．実際，現実におきる機械スケジューリング問題はすべて近似解法によって解決されているといって過言でない．ここでいう近似解法は“常に最適解を与える保証はないが，多くの場合最適に近い実行可能解を比較的少ない計算で与える方法”を意味する．

1つの問題に対して無数の近似解法が考えられるが，これらのほとんどは極端にいえば，単なる思いつきに過ぎないいわゆる発見的方法にもとづく．したがって近似解法の最大の欠点はそれによって得られる近似解の誤差の程度が明らかでないことである．この意味において近似解法の性能を評価するため，従来多数の問題例を実際に解いてみるという数値実験が行なわれている．数値実験による評価法は2通りある．1つは複数個の近似解法を同じ問題例に適用して結果の優劣を調べる相対的な方法である．もう1つは近似解を求めるとともに何らかの方法で厳密解をも求めてその誤差を計算する絶対評価法である．これらの実験は種々の情報を提供する点で有利である反面，いくつかの欠点をもつ．相対評価法は必ずしも比較された解法の中で最良のものが実際にすぐれていることを保証しない．絶対評価法は小規模の問題例にしか適用できない．さらにこれらによって得られた評価は厳密にいえば調べた問題例にしか適用できない．

これらの欠点を補うため，性能保証(performance guarantee)された近似解法が最近盛んに論じられている．その1つが ε 近似(ε approximation) である．近似値の相対誤差が常にある値 ε

きせ　ひろし　京都工芸繊維大学　工芸学部

より大きくないことが保証されている近似アルゴリズムを ε 近似アルゴリズムという．もう1つは全多項式時間近似 (fully polynomial time approximation) である．任意の $\varepsilon(>0)$ について問題の規模と $(1/\varepsilon)$ の多項式程度の計算で収束する ε 近似アルゴリズムを全多項式時間近似アルゴリズムという．

ここではまず，NP 完全理論によって機械スケジューリング問題の解ける限界を示す．次に若干の機械スケジューリング問題に対する ε 近似アルゴリズムと全多項式時間近似アルゴリズムを論議する．ε 近似アルゴリズムについてはそれらについて行なわれた数値実験の結果をも示す．また，両近似法が適用できる限界と今後の展望についても若干触れることにする．

2. 問題の複雑性

機械スケジューリング問題の解集合は有限である．よってすべての解を列挙すれば問題は解決する．たとえば，n 個の仕事を1台の機械で処理する"1機械"問題では n！通りの順序を列挙すればよい．もちろん，このような方法が実際上不可能なことは 10！ がどれほどになるかを見ればよい．しかし以下で示すように実は大方の機械スケジューリング問題はこういう列挙にもとづく方法でしか解けないのである．

問題を解くアルゴリズムの計算量を測る尺度として問題の規模をとる．通常，計算機を使うから問題の規模はその問題を規定する入力データのビット数で表わされる．たとえば上記の1機械問題が各仕事 i の処理時間 p_i のデータだけで規定されるならば，問題の規模は $I=\sum_{i=1}^{n}\log p_i$ となる．実際には少し大雑把に数えてデータの代表値とその個数で表わすこともある．上の例では，

$I=n\log p_{\max}$, $p_{\max}=\max_i p_i$ である．

機械スケジューリング問題のみならず，よく知られた多くの組合せ最適化問題を包含するクラス NP と言われるクラスがある．クラス NP に属する問題は高々問題規模 I の指数関数 $\exp(a_1 I)$ 程度の計算量で解かれることが知られている (a_1 は定数)．特に NP の中で多項式 $a_2 I^b$ ($a_2>0$ と b は定数)程度の計算時間で収束するアルゴリズム(多項式時間アルゴリズムという) で解ける問題のクラスをクラス P という．クラス P に属する問題を効率よく解ける(tractable)問題，NP にあって P にない問題を効率よく解けない (intractable) 問題と区別することは直観的にうなずける．しかしながら問題が効率よく解けない (P にない) ことをどうして証明するか．実は多項式時間アルゴリズムが絶対ないことが証明された問題はまだ見出されていない．ところがある問題が効率よく解けるならば，他のすべての問題も効率よく解けること，すなわち $NP=P$ が証明できる問題がある．このような NP の中で最も難しいと見られる問題を NP 完全問題という．これまでに多数の NP 完全問題が見出されている (文献 [5] に網羅されている)． それらすべてが効率よく解ける問題になるとは考え難いところから，現在では NP 完全問題は効率よく解けないとされている．

大方の機械スケジューリング問題が NP 完全であることを示すためには最も単純な状況を想定した次の並列機械スケジューリング問題 (parallel-machine scheduling problem) を考えれば十分である．

並列機械スケジューリング問題　n 種の仕事 $i=1,2,\cdots,n$ を処理するため $m(\geq 1)$ 台の機械がある．各仕事 i は任意の機械で1度だけ処理されれば完了する．このとき課せられた制約条件を満たし，与えられた評価関数の最適値を達成するスケジューリングを見出せ．制約条件として次の2つを考える．

半順序　いくつかの仕事の対 (i,j) に対して i が完了する前に j を始めてはいけない．

最早開始時刻(準備時間)．各仕事 i を与えられた時刻 r_i より早く開始してはいけない．

評価関数として次のようなものが考えられる．

表 1 NP 完全な並列機械スケジューリング問題

評価関数	機械台数$^{(f)}$	制約条件
$C_{\max}$	$m\geq 2$	なし
$\sum c_i$	$m\geq 1$	半順序または最早開始時刻
$\sum w_i c_i$	$m\geq 2$	なし
$L_{\max}$	$m\geq 1$	最早開始時刻
$\sum T_i$	$m\geq 1$	半順序または最早開始時刻
$\sum w_i T_i$	$m\geq 1$	なし
$\sum U_i$	$m\geq 1$	半順序または最早開始時刻
$\sum w_i U_i$	$m\geq 1$	なし

(f) $m\geq k$ は k 以上の任意の m について成立つことを示す.

いずれも最小値を達成するスケジュールが最適である. なお, 仕事 i の完了時刻を c_i と記す. また各仕事 i には納期 d_i と重み w_i が与えられている.

総完了時刻 $c_{\max}=\max_i c_i$

滞留時間 $\sum c_i=\sum_i c_i$, 重みつき滞留時間 $\sum w_i c_i$,

最大納期遅れ $L_{\max}=\max_i(c_i-d_i)$,

遅れ和 $\sum T_i=\sum_i \max(0, c_i-d_i)$,

重みつき遅れ和 $\sum w_i T_i$

遅れ仕事数 $\sum U_i=\sum_i U_i$, ただし $c_i>d_i$ のとき $U_i=1$, それ以外は $U_i=0$,

重みつき遅れ仕事数 $\sum w_i U_i$.

表 1 は NP 完全である並列機械スケジューリング問題の例を示す. これらの例を特別な場合として含むいかなる問題もまた NP 完全であるから, 現実におきる大方の問題が NP 完全であることは自明であろう. なお, NP 完全を含めたスケジューリング問題の複雑性については文献 [5,7] が詳しい.

3. ε 近似アルゴリズム

多項式時間アルゴリズムが問題 Q の ε 近似アルゴリズムである必要十分条件は Q の任意の問題例 $Q(I)$ にそのアルゴリズムを適用したとき,

$$E(Q(I))\equiv|F^*(Q(I))-\hat{F}(Q(I))|/F^*(Q(I))\leq\varepsilon$$

を満たす相対誤差 $E(Q(I))$ が得られることである. ここで F^* は最適値, $\hat{F}$ は近似値を表わす. なお, $E(Q(I))=\varepsilon$ を満たす問題例 $Q(I)$ があれば, ε は到達可能な最良の上限であるいわれる. ここでは次の 1 機械問題 Q に対する ε 近似アルゴリズムを論議する.

問題 Q n 仕事 $i=1,2,\cdots,n$ を 1 機械で処理する. 各仕事 i の最早開始時刻を r_i, 処理時間を p_i, また完了した仕事を発注者に返送する時間を s_i とするとき, 最大リード・タイム

$$L_{\max}\equiv\max_i(c_i+s_i)$$

を最小にする仕事の処理順序(スケジュール)を求めよ. ただし, c_i は仕事 i の完了時刻である.

問題 Q は最早開始時刻制約をともなう 1 機械最大納期遅れ問題と等価である ($s_i=-d_i$ とすればよい). よって表 1 に示すように Q は NP 完全である. 次の 2 つのアルゴリズムは古くから知られている.

アルゴリズム J(Jackson [10])

返送時間の非増大順に処理する. これによって得られる順序を $\zeta=(\zeta_1,\zeta_2,\cdots,\zeta_n)$ とすると, $s_{\zeta_1}\geq s_{\zeta_2}\geq\cdots\geq s_{\zeta_n}$. ただし ζ_k は k 番目に処理される仕事を表わす.

アルゴリズム S(Schrage [16])

最早開始時刻が最小の仕事から始める. 処理中の仕事が完了するごとに, その時刻より大きくない最早開始時刻をもつ仕事の中から最大の返送時間をもつ仕事を次に処理する.

ところで問題 Q は r_i と s_i に関して対称である. $R=(r_1,r_2,\cdots,r_n)$, $P=(p_1,p_2,\cdots,p_n)$, $S=(s_1,s_2,\cdots,s_n)$ とするとき, 最早開始時刻, 処理時間, 返送時間の集合がそれぞれ R,P,S である問題例 ($Q(R,P,S)$ と記す) と S,P,R をそれぞれ最早開始時刻, 処理時間, 返送時間とする問題例 $Q(S,P,R)$ は互いに逆であるという. また順序 $\pi=(\pi_1,\pi_2,\cdots,\pi_n)$ (π_k は k 番目に処理される仕事) と順序 $\overleftarrow{\pi}=(\overleftarrow{\pi}_1,\overleftarrow{\pi}_2,\cdots,\overleftarrow{\pi}_n)$ は, $\overleftarrow{\pi}_k=\pi_{n-k+1}$, $1\leq k\leq n$ を満たすならば, 互いに逆であるという. 次の定理が成立つ.

表 2　1機械問題に対する ε 近似アルゴリズム

アルゴリズム	$J,\ \bar{J}$	mJ	$S,\ \bar{S}$	mS
ε	$1-(1/\sum p_i)$	$1-(2/\sum p_i)$	$1-(3/(\sum p_i+1))$	$1-(5/(\sum p_i+2))$
計算量	$0(n \log n)$			

定理 1［12］　互いに逆である2つの問題例に対するそれぞれのスケジュールが互いに逆ならば，それらの最大リード・タイムは等しい．

注意すべきは互いに逆な2つの問題例は定理1の意味で等価であるが，1つの近似アルゴリズムをこれらに適用したとき，必ずしも同じ結果が得られるとは限らないという点で等価でない．このことから次の4種の新しいアルゴリズムが得られる．

アルゴリズム $\bar{J}$．元の問題の逆問題にアルゴリズム J を適用する．得られた順序の逆順序を元の問題のスケジュールとする．

アルゴリズム $\bar{S}$．元の問題の逆問題にアルゴリズム S を適用する．得られた順序の逆問題を元の問題のスケジュールとする．

アルゴリズム mJ．アルゴリズム J と $\bar{J}$ の両方を適用し，得られた結果のよいほうを選ぶ．

アルゴリズム mS．アルゴリズム S と $\bar{S}$ の両方を適用し，得られた結果のよいほうを選ぶ．

上記の6アルゴリズムの性能を示す前に次の事実を知ることは興味深い．

定理 2［12］　Q の任意の問題例についての任意のスケジュールの相対誤差が $2-(2/\sum p_i)$ を越えることはない．またこの上限に到達する問題例とスケジュールの組がある．

定理 3［12］　上記の各アルゴリズムの性能を表2に示す．また表中の各 ε はすべて相対誤差に対する到達可能な最良の上限である．ただし，$\sum p_i = \sum_i p_i$

定理1と定理2から上記アルゴリズムのいずれかを使えば，無作為にスケジュールを選んだ場合より相対誤差を半分以下に減らせることがわかる．また表2は6種のアルゴリズムの性能は ε と計算量の程度に関する限り，あまり変わらないことを示している．しかし相対誤差が ε に到達する例はきわめて稀で実用上より重要と思われる平均的な相対誤差のふるまいはどうか．これを調べるため数値実験をした．

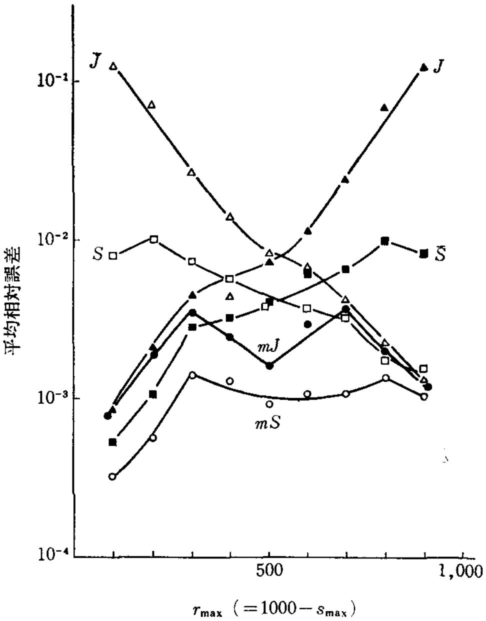

図 1　実験結果

図1は各アルゴリズムの平均相対誤差(100問題例についての平均値)に対する最早開始時刻と返送時間の影響を示す．ただし，$r_{\max}+s_{\max}=1000$ を満たす〔$0, r_{\max}$〕,〔$0, s_{\max}$〕の範囲から一様乱数によって，それぞれ r_i, s_i を選んだ．また，仕事数 $n=20$ とし，〔$1, p_{\max}$〕, $p_{\max}=25$ から一様乱数によって p_i を生成した．この結果から，第1に各アルゴリズムの平均相対誤差は $\varepsilon \fallingdotseq 1$ よりいちじるしく小さい点が注目される(なお，$r_{\max}=0$ での $\bar{J}$ または $s_{\max}=0$ での J に対する平均相対誤差は無作為に選んだ順序の平均相対誤差にほぼ等しいと考えられる)．第2に注目すべき点は平均相対誤差についてはアルゴリズムによっていちじるしい差が認められることである．特に，アルゴリズム mS はほとんど最適解と変わらない近似解を与えているように見える．

ところで，ここでとりあげた1機械問題 Q を厳

密に解くための分枝限定法アルゴリズム[1,2,13,14,15]が提案されているが，いずれもが行なわれた計算実験によると80以上の仕事数の問題例を実際上解くことができないようである．

他の機械スケジューリング問題に対する ε 近似アルゴリズムについては文献[3,6,7]が詳しい．

4. 全多項式時間近似アルゴリズム

任意の ε について ε 近似を満たし，かつ計算量を問題の規模と $(1/\varepsilon)$ の両方についての多項式程度で抑えることができるアルゴリズムを全多項式時間近似アルゴリズムという．ここでは1機械重みつき遅れ仕事数問題(仕事 $i=1,2,\cdots,n$ を1機械で処理するとき，重みつき遅れ仕事数 $\sum w_iU_i$ を最小にするスケジュールを求める）を考える．表1に示したようにこの問題は NP 完全である．この問題では納期に間に合う仕事の集合の後に納期に遅れた仕事の集合が処理されるとしてよい．また，納期遅れのないスケジュールが存在する必要十分条件は納期の非減少順に対応するスケジュールに納期遅れがないことであるというよく知られた事実[10]から，間に合う仕事は常に納期順に順序づけられているとする．よって $d_1\leq d_2\leq\cdots\leq d_n$ とすれば，上の問題は0-1整数計画問題 Q として定式化できる．

問題 Q $\max \sum_{i=1}^{n} w_ix_i$

$$\text{s.t. } c_k=\sum_{i=1}^{k} p_ix_i\leq d_i,\ 1\leq k\leq n,$$
$$x_i=0 \text{ or } 1,\ 1\leq i\leq n.$$

明らかに $x_k=1$ は仕事 k が納期に間に合うことを，$x_k=0$ は遅れることを意味する．なお，問題 Q では $w_i, p_i>0$, $p_i\leq d_i$, $1\leq i\leq n$ としてよい．

問題 Q は動的計画法でいうところの最適性の原理を満たす．すなわち，ある k $(1\leq k\leq n)$ に対する部分解 $x_i=y_i$, $1\leq i\leq k$ を満たす Q の最適解 $x_i=y_i$, $1\leq i\leq n$ があるならば，$x_i=y_i$, $k\leq i\leq n$ は対応する次の部分問題 $Q_k(w,t)$ の最適解である．

$Q_k(w,t)$ $\max(\sum_{i=k+1}^{n} w_ix_i)+w$

$$\text{s.t. } \sum_{i=k+1}^{j} p_ix_i\leq(d_j-t),\ k<j\leq n$$
$$x_i=0 \text{ or } 1,\ k<i\leq n.$$

ただし，$w=\sum^{k}_{i=1}w_iy_i$, $t=\sum^{k}_{i=1}p_iy_i$. よって $\{Q_k(w,t)\}$, $k=1,2,\cdots,n$ を次々と生成することによって原問題 Q が解ける．最大の w をもつ $Q_n(w,t)$ の最適値が Q の最適値である．しかしながら，この多段決定過程では k 段目の1つの部分問題 $Q_k(w,t)$ は $(k+1)$ 段目の2つの部分問題 $Q_{k+1}(w,t)$ と $Q_{k+1}(w+w_{k+1},\ t+p_{k+1})$ に分解されるので，全体として生成される部分問題の数は最悪の場合，

$\sum_{i=0}^{n-1}2^i=2^n-1$ 個となる．したがって動的計画法のアルゴリズムをそのまま適用すると，問題規模の指数関数の計算時間が必要となる．

全多項式近似解法は動的計画法で生成される部分問題の数を減らすために次の優越規則（dominancerelation）を用いる．

優越規則 k 段目の2つの部分問題 $Q_k(w,t)$ と $Q_k(w',t')$ が $w\geq w'$, $t\leq t'$ を満たすならば，$Q_k(w,t)$ は $Q_k(w',t')$ を優越するという．明らかに $Q_k(w',t')$ 以外に Q の最適値をもつ k 段目の部分問題があるので，$Q_k(w',t')$ を生成する必要がない．この優越規則の適用は異なる値の w をもつ $Q_k(w,t)$ のみが実際に生成されることを意味する．

問題 Q の最適値 w^* の下限を B とする．たとえば，$B=\max_i w_i$ とすることができる．ここで，問題 Q の代わりに重みが，

$\hat{w}_i=w_i-\text{rem}(w_i,B\varepsilon/n)\leq w_i, 1\leq i\leq n$, である別の問題 $\hat{Q}$ を考える(p_i, d_i は変わらない)．ただし，$\text{rem}(a,b)=a-\lfloor a/b\rfloor b$($\lfloor x\rfloor$ は x を越えない最大の整数)．処理時間，納期は変わらないので $\hat{Q}$ の最適解は元の問題 Q の実行可能な近似解である．また $\text{rem}(w_i,\ B\varepsilon/n)<B\varepsilon/n$ から $\sum_{i=1}^{n}(w_i-\hat{w}_i)=\sum_{i=1}^{n}\text{rem}(w_i,\ B\varepsilon/n)<B\varepsilon\leq w^*\varepsilon$. さらに，$\sum_{i=1}^{n}(w_i-\hat{w}_i)\geq\sum_{i=1}^{n}(w_i-\hat{w}_i)\,x_i^*\geq w^*-\hat{w}$. ただし，$x_i^*$ は Q の最適解，w^*, $\hat{w}$ はそれぞれ Q と $\hat{Q}$ の最適値である．以上から，$(w^*-\hat{w})/w^*\leq\varepsilon$ となるので，Q の代わりに $\hat{Q}$ を解けば，相対誤差

が ε 以下の Q の近似解が得られる．このように Q の代わりに丸められた重み $\hat{w}_i$ をもつ $\hat{Q}$ を解く手法を丸め法(rounding)という．

$\hat{Q}$ においては同じ値の重みが多くなることからその部分問題の多くが優越規則で除外されることが予想できる．実際，生成されるべき部分問題の数が$O(2^n)$から$O(n^3/\varepsilon)$に減られることを次に示そう．$\hat{Q}$の代わりに重みが$\tilde{w}_i=\lfloor(w_in)/B\varepsilon\rfloor=(\hat{w}_in)/(B\varepsilon)$, $1\leq i\leq n$ で与えられるもう1つの問題$\tilde{Q}$を考える．$\hat{Q}$ と $\tilde{Q}$ において異なる値の重みの数は等しいから，生成すべき部分問題の数は同じである．他方，$w_i\leq B=\max_i w_i$ から $\tilde{w}_i\leq\lfloor n/\varepsilon\rfloor$ となるから，$|\{Q_k(w,t)\}|\leq 1+\sum_{i=1}^{k}\tilde{w}_i\leq 1+\lfloor n/\varepsilon\rfloor$($|X|$は集合 X の要素の数)．よって $\sum_{k=0}^{n}|\{\tilde{Q}_k(w,t)\}|\leq n+\sum_{k=0}^{n}k\lfloor n/\varepsilon\rfloor=O(n^3/\varepsilon)$ となるので，丸め法にもとづく近似アルゴリズムの計算量は $O(n^3/\varepsilon)$ となる．

表3の(p_i, d_i, w_i), $1\leq i\leq 4$ で表わされる問題例を $\varepsilon=0.1$ で解こう．$B=\max_i w_i=200$とすると，丸められた重み $\hat{w}_i$ は表3の右欄のようになる．表4は各段階 k において生成された部分問題 $\{\hat{Q}_k(w,t)\}$ を2項組 (w,t) で表わす．この結果，近似解は$\hat{x}_1=\hat{x}_2=1$, $\hat{x}_3=\hat{x}_4=0$, $\hat{w}=300$となる．他方，π の問題の最適解は $x_1{}^*=x_3{}^*=1$, $x_2{}^*=x_4{}^*=0$, $w^*=304.9$ である．よって $(w^*-\hat{w})/w^*=4.9/304.9<0.02<0.1=\varepsilon$.

表 3　1機械重みつき遅れ仕事数問題の例

仕事 i	処理時間p_i	納期 d_i	重み w_i	重み $\hat{w}_i$
1	2	2	200	200
2	1	3	100	100
3	3	5	104.9	100
4	2	5	5	5

表 4　丸め法による近似解の計算

k	生成された部分問題$\{Q_k(w,t)\}$
1	(0, 0)
2	(0, 0), (200, 2)
3	(0, 0), (200, 2), (100, 1), (300, 3)
4	(0, 0), (200, 2), (100, 1), (300, 3)*

以上が丸め法による全多項式時間近似の一例である．この他に，区間分割法，分離法にもとづく全多項式時間近似が提案されている[18]．また，全多項式時間近似が適用できる他の機械スケジューリング問題も報告されている[8, 17].

5. まとめ

NP 完全問題は擬似多項式時間アルゴリズム(データの数については多項式，データの大きさについては指数関数(本文2. を参照)）で解ける弱 *NP* 完全問題とそうでない強 *NP* 完全問題に分けられる [4]．たとえば，最早開始時刻制約をともなう1機械最大納期遅れ問題は強 *NP* 完全で，1機械重みつき遅れ仕事数問題は$O(n\sum p_i)$で解ける弱 *NP* 完全である．強 *NP* 完全問題を任意の ε について ε 近似することはまた *NP* 完全であること[19]が知られているので，全多項式時間近似できる問題の範囲は限定される．強 *NP* 完全問題を任意の ε について ε 近似できる1つの方法として近似分枝限定法がある．この方法は，最悪の場合，問題規模の指数関数程度の計算量を要する [9] ものの，実用上期待できる汎用近似解法の1つと考えられる．

他方，近似アルゴリズムを，それを使っておきる最悪の場合で評価する ε 近似はすべての問題に適用でき，かつ現在盛んに行なわれている．しかしながら，本文の例でも見られたように，ε 近似はしばしばアルゴリズムを過少評価する傾向がある．この難点を解消するための2つの方向が考えられる．1つは ε を単に数値で表現するのでなくて，多くの問題に関する情報(入力パラメータ等)を含めた形で表わして，与えられた問題例の特長を反映させる方向である．もう1つは近似アルゴリズムが最適解を与える程度を確率的に解析する方向[11]である．これらの研究は現在，端緒についたばかりであるが，今後大いに発展するものと

期待される.

参考文献

[1] K. R. Baker. Z-S Su (1974), "Sequencing with Due-Dates and Early Start Times to Minimize Maximum Tardiness," *Naval Res. Logist. Quart.* **21**, 171-176.

[2] P. Bratley, M. Florian, P. Robillard (1973), "On Sequencing with Earliest Start and Due-Dates with Application to Computing Bounds for the $(n/m/G/F_{max})$ Problem," *ibid*, **20**, 57-67.

[3] M. R. Garey, D. S. Johnson (1976), "Approximation Algorithms for Combinatorial Problems: an Annoted Bibliography," Algorithm and Complexity: New Direction and Recent Results (J. F. Traub, ed)., Academic Press, New York, 41-52.

[4] ——, ——(1978), "Strong NP-Completeness Results: Motivation, Examples and Implication," *J. ACM*. **25**, 499-508.

[5] ——, ——(1979), "Computer and Intractability: A Guide to the Theory of NP-Completeness", W. H. Freeman and Company, San Francisco.

[6] M. R. Garey, R. L. Graham, D. S. Johnson, (1978), "Performance Guarantees for Scheduling Algorithms," *Operations Res.* **26**, 3-21.

[7] R. L. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, (1979), "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey," *Ann. Discrete Math.*, North-Holland, **5**, 287-326.

[8] E. Horowitz, S. Sahni (1976), "Exact and Approximate Algorithms for Scheduling Nonidentical Processors," *J. ACM*. **23**, 317 327.

[9] Ibaraki (1976), "Computational Efficiency of Approximate Branch-and-Bound Algorithms," *Math. Operations Res.* **1**, 287-297.

[10] J. R. Jackson (1955), "Scheduling a Production Line to Minimize Maximum Tardiness," Research Report No. 43, Management Sci. Res. Project, Univ. California at Los Angeles.

[11] R. M. Karp (1977), Probablistic Analysis of Some Combinatorial Search Algorithms," Algorithms and Complexity: New Derection and Recent Results (J. F. Traub, ed.), Academic Press, New York, 1-19.

[12] H. Kise, T. Ibaraki, H. Mine (1979), "Performance Analysis of Six Approximation Algorithms for the One-Machine Maximum Lateness Scheduling Problem with Ready Times," 本学会論文集, **22**, 205-224.

[13] B. J. Lageweg, J. K. Lenstra, A. H. G. Rinnooy Kan (1976), "Minimizing Maximum Lateness on One Machine: Computational Experience and Some Applications," *Statistica Nearlandica* **30**, 25-41.

[14] G. Mcmahon, M. Florian (1975), "On Scheduling with Ready Times and Due Dates to Minimize Maximum Lateness," *Operations Res.* 23, 475-482.

[15] 三根久, 茨木俊秀, 木瀬洋 (1974), "ある種のスケジューリング問題に対するアルゴリズム", 経営科学, **18**, 23-37.

[16] L. Schrage (1971)., "Obtaining Optimal Solutions to Resources Constrained Network Scheduling Problems," Unpublished Manuscript.

[17] S. Sahni (1976), "Algorithms for Scheduling Independent Tasks," *J. ACM*. **23**, 116-127.

[18] S. Sahni (1977), "General Techniques for Combinatorial Approximation," *Operations Res.* **25**, 920-936.

[19] S. Sahni, E. Horowitz (1978), "Combinatorial Problems: Reducibility and Approximation," *Operations Res.* **26**, 718-758.