時間制約付きプロジェクト計画法*

藤井 進**・森田 浩**・金田 健一***・正岡 良規****

Multi-Project Planning with Time Constraints*

Susumu FUJII**, Hiroshi MORITA**, Ken-ichi KANEDA*** and Yoshinori MASAOKA****

We consider a multi-project scheduling problem with time constraints on resources, which appear in the planning of new projects that need to use the same resources as other projects under execution. We seek the optimal sequence of activities and the shortest completion time for all projects taken together. We propose a solution procedure using a branch and bound method for small size problems, where the bounding operations are established by a lower bound as well as an equivalence relation between the sequences of activities. An approximate solution by heuristics is required because of the high computational complexity. The approximate solutions obtained by eight heuristic rules are compared with the shortest completion times by using randomly generated simulating problems. We conclude that an earliest finishing time order rule provides effective heuristic rules for multi-project planning with time constraints on resources; however, a latest finishing time order rule is effective for a large-scale project.

1. はじめに

大規模なプロジェクトを円滑に遂行するための計画手法として, PERT(Program Evaluation and Review Technique) や CPM(Critical Path Method) がよく用いられている. これらの手法は, プロジェクトを構成する作業間の先行関係を保ちながら最も早くプロジェクトを完了させるように, 作業順序を計画するものである. しかし, 作業を実施するに当たって必要とされる人員, 資材, 設備などの資源量やそれらを使用できる時間帯などの条件は考慮に入れられておらず, 任意の資源量をいつでも使

用することができるものとして計画がなされている.

また, 同種のプロジェクトが並列的に進行している場合, 資源量の制約などにより, 同じ作業を同時に実行できるとは限らない. このとき各時間における使用可能な資源量を考慮に入れ, 各作業の実施順序を決定するためのプロジェクト計画手法の開発が必要となる. 従来の資源制約つきプロジェクト計画法は, プロジェクト数, 目的関数, 資源や作業の種類などによって分類される. 目的関数には複数プロジェクトの完了時間の重みつき和の最小化問題⁴⁾, プロジェクト遂行時間最小化問題¹⁾, プロジェクト総費用最小化問題⁶⁾がある. 人員や設備などは再利用可能な資源であり, 多くの研究で扱われているが, 一度使ったら再び利用することのできない資金や燃料などの再利用不可能な資源⁷⁾も存在する. また, より多くの資源量を使用すれば作業時間を短縮することが可能な場合もあり, 資源量と作業時間の間の関係を設けた定式化⁵⁾も行われている. このようなプロジェクト計画問題は, 資源の割り当て順序あるいは作業の処理順序を決定する組合せ最適化問題として定式化され, 分枝限定法や陰の列挙法といった最適手法や経験則に基づいた近似解法などが用いられている.

* 原稿受付 1995年2月28日

** 神戸大学 工学部 Faculty of Engineering, Kobe University; 1-1, Rokkodai-cho, Nada ward, Kobe city, Hyogo 657, JAPAN

*** 花王(株) 第一生産技術開発研究所 Chemical Process Development Institute, Kao Corporations; 1334 Minato, Wakayama city, Wakayama 640, JAPAN

**** (株)東芝 研究開発センター Research and Development Center, Toshiba Corporations; 70 Yanagimachi, Saiwai ward, Kawasaki city, Kanagawa 210, JAPAN

Key Words: multi-project planning, time constraints, branch and bound method, equivalence relation between activity sequences, heuristic rule.

これらのほとんどは、主として有限の資源量をいかに配分するかという点から研究されており、作業が任意の時間に実行できるという前提のもとで行われている。本論文では、使用できる資源量の制約により、作業が実施できる時間帯と実施できない時間帯が存在する場合を考える。すでに、あるプロジェクトが計画され進行中のとき、新たなプロジェクトを計画するには、残存している資源や時間を使って行わなければならない、作業の実施可能な時間帯が限られてくる。

まず、2.で対象とするプロジェクト計画問題の定義を行い、資源量に時間制約のあるときの完了時間最小化問題を定式化する。分枝限定法による解法を3.で与え、より大規模なプロジェクトに適用するための経験則に対する比較検証の数値実験を行った結果を4.で示している。

2. 資源量制約のあるプロジェクト計画問題

大規模プロジェクトにおいては、各作業はそれを専門とする職能集団や特殊機器によって実行される場合が多く、これらが他のプロジェクトに従事しているときには新たなプロジェクトに従事することは不可能である。このとき、すでに確定しているプロジェクトの計画を変更して新たなプロジェクトに従事させる場合と、すでに確定しているプロジェクトに従事していない時間帯の中で新たなプロジェクトに従事させる場合がある。前者の場合はすでに計画されたプロジェクトを、まだ実施されていない作業からなる新たなプロジェクトとみて、複数プロジェクトの同時計画を行えばよい。本論文では、後者の場合の作業を実施することのできる時間帯が限られているときの新規プロジェクトの計画問題を考える。

2.1 対象とするプロジェクト

Fig.1に示すような m 個の同種のプロジェクト $P_i, i=1, 2, \dots, m$ の実施計画を考える。各プロジェクトにおける作業間の先行関係は同一であり、同じネットワークで表現されるものとする。各プロジェクトはそれぞれ n 個の作業から構成され、 i 番目のプロジェクトの j 番目の作業 a_{ij} の作業時間 d_{ij} と実施に必要な資源量 q_{ij} は所与とする。各プロジェクトの j 番目の作業の集まりを第 j 作業群 A_j と呼ぶ。資源の競合は同一の作業群でのみ生じ、一つのプロジェクト内では生じないものとする。この仮定は各作業を担当する職能集団は他の作業を担当しないということに対応しており、作業群ごとに実施に必要な資源の種類が異なるものとしている。したがって、Fig.1に示しているように、作業群 A_j に対してそれぞれ資源の種類 R_j が一意に対応する。ただし、同一の作業群内の作業の作業時間や必要な資源量は等しくなくてもよい。

各作業は時間分割が可能であるとし、連続的に作業することができないときには作業を一時中断して次の作業

Fig. 1 Multi-project planning model

Fig. 2 Example for available amount of resource $\Gamma_j(k)$ and time windows $\Omega_j(q)$

可能な時間帯で再開することができるものとする。時刻 k における資源 R_j の使用可能量を $\Gamma_j(k)$ とすると、第 j 作業群で資源量 q を必要とする作業の実施可能時間帯は

$$\Omega_j(q) = \{k | \Gamma_j(k) \geq q\} \quad (1)$$

と表わされ、作業 a_{ij} の実施可能時間帯 Ω_{ij} は $\Omega_j(q_{ij})$ となる。これらの数値例をFig.2に示す。

以上の条件のもとですべてのプロジェクトの完了する時刻を最早にするためのプロジェクト計画を求めること

を目的とする。ここでのプロジェクト計画とは、すべての作業群に対して、それぞれの作業群内での作業の順序を決定することである。各作業群は m 個の作業を含んでいるので、その作業順序は $m!$ 通りある。 n 個の作業群すべてについての作業順序は $(m!)^n$ 通りにもなり、この中から最適なプロジェクト計画を見つける問題は m を定数とみても NP 完全となる²⁾。

2.2 記号の定義

1 次元実数空間上の閉区間 $X^k, k=1,2,\dots,K$ を要素にもつ集合 X に対して、最小値 $\min X$ 、最大値 $\max X$ および大きさ $\|X\|$ を以下のように定義する。

$$\min X = \min \{x | x \in X^k, k=1,2,\dots,K\} \quad (2)$$

$$\max X = \max \{x | x \in X^k, k=1,2,\dots,K\} \quad (3)$$

$$\|X\| = \sum_{k=1}^K |X^k| \quad (4)$$

さらに、閉区間 $Y^h, h=1,2,\dots,H$ を要素にもつ集合 Y との積演算を

$$X \otimes Y = \bigcup_{k,h} [X^k \cap Y^h] \quad (5)$$

と定義する。

2.3 作業列にしたがった作業の割り当て

作業 a_{ij} の実施を開始することのできる時刻 TS_{ij} 、作業 a_{ij} の最早開始時刻 ES_{ij} および作業 a_{ij} の最早終了時刻 EF_{ij} は次式で与えられる。

$$ES_{ij} = \max \{TS_{ij}, \min \Omega_j(q_{ij})\} \quad (6)$$

$$EF_{ij} = \min \{t | \|[ES_{ij}, t] \otimes \Omega_j(q_{ij})\| \geq d_{ij}\} \quad (7)$$

$$TS_{ij} = \max_k EF_{ik} \quad (8)$$

ただし、(8) 式は作業 a_{ij} に先行する作業 a_{ik} の中の最早終了時刻の最大値であり、 $TS_{i1} = 0$ である。

また、プロジェクト全体を時刻 f までに完了させるものとしたとき、作業 a_{ij} を終了しておかなければならない時刻 TF_{ij} 、作業 a_{ij} の最遅開始時刻 LS_{ij} および作業 a_{ij} の最遅終了時刻 LF_{ij} は次式で与えられる。

$$LF_{ij} = \min \{TF_{ij}, \max \Omega_j(q_{ij})\} \quad (9)$$

$$LS_{ij} = \max \{t | \|[t, LF_{ij}] \otimes \Omega_j(q_{ij})\| \geq d_{ij}\} \quad (10)$$

$$TF_{ij} = \min_k LS_{ik} \quad (11)$$

ただし、(11) 式は作業 a_{ij} に後行する作業 a_{ik} の中の最遅開始時刻の最小値であり、 $TF_{in} = f$ である。

作業群 A_j の作業列 $\gamma = \{a_{i1j}, a_{i2j}, \dots, a_{imj}\}$ が与えられたとき、この順序にしたがって作業を実施可能な時間帯へ割り当てるには、以下のアルゴリズムによって使用

Fig. 3 Branch and bound method

可能量 $\Gamma_j(k)$ を更新していけばよい。

ステップ 1 作業列 γ の最初の作業を a_{ij} として、その実施可能時間帯 Ω_{ij} を求める。

ステップ 2 作業 a_{ij} の最早開始時刻 ES_{ij} と最早終了時刻 EF_{ij} を計算し、作業 a_{ij} の実施可能時間帯 W を求める。

$$W = \Omega_{ij} \otimes [ES_{ij}, EF_{ij}] \quad (12)$$

ステップ 3 作業 a_{ij} を実施可能時間帯 W へ割り当て、使用可能量 $\Gamma_j(k)$ から、使用した資源量 q_{ij} だけ減らして $\Gamma_j(k)$ を更新する。

$$\Gamma_j(k) = \begin{cases} \Gamma_j(k) - q_{ij}, & k \in W \\ \Gamma_j(k), & k \notin W \end{cases} \quad (13)$$

ステップ 4 作業列 γ から作業 a_{ij} を削除する。 $\gamma = \emptyset$ ならば終了、そうでなければステップ 1 へもどる。

3. 分枝限定法による解法

プロジェクトが最も早く完了するための作業の割り当てを決定するプロジェクト計画問題を考える。この問題は作業群内の作業の割り当て順序をそれぞれ決定する問題であり、**3.**では分枝限定法による解法を示す。

探索木は、深さ j の節点に作業群 A_j に含まれる作業の割り当て順序を表す作業列 γ を対応させて、Fig. 3 のように構成する。分枝の際に生成される作業列の個数は $m!$ 個になるが、その生成順序は各作業の最早終了時刻を辞書式順序に並べたものとする。最早終了時刻順による割り当て方法は、資源制約のあるプロジェクト計画に対して、比較的有効であることが示されている³⁾。本論文では、分枝数が極めて多くなるため、効率的な限定操作を行う必要があり、下界値による限定のほかには節点間の同値関係を導入して限定操作を行う。

Fig. 4 Equivalence relation between sequences of activities

3.1 下界値による限定操作

深さ j における節点の下界値 LB は次のようにして求めている。まず、第 1 作業群 A_1 から第 $j-1$ 作業群 A_{j-1} まではすでに割り当て順序が決定しているの、それにしたがって割り当てを行う。つぎに、第 j 作業群 A_j から第 n 作業群 A_n までは、他の作業との競合は無視して最も都合のよい状態で作業を実施したときのプロジェクト全体の完了時間を求め、これを下界値 LB とする。

このとき、下界値 LB が現段階での完了時刻の上界値 UB より大きければ、節点をこれ以上分枝しても上界値 UB が更新されることはないの、この節点は終端される。

さらに、下界値 LB をプロジェクトの完了時刻 f として各作業 a_{ij} の最遅終了時刻 LF_{ij} を求めておくと、節点の下界を求める途中で作業 a_{ij} の終了時刻が LF_{ij} より遅くなれば、この節点を終端させることもできる。

3.2 同値関係による限定操作

同一の節点から分枝された節点の中で、作業を実施する順序が異なっても、各作業の開始時刻と終了時刻が全く同じとなることがある。これらの節点を同値関係にあると定義する。Fig. 4 に示す例で第 j 作業群の四つの作業である a_{1j} から a_{4j} の作業群を考える。各作業の必要資源量、開始可能時間および作業時間は Table 1 で与えられている。この場合、どんな作業列においても、部分作業群 $\{a_{1j}, a_{2j}\}$ の作業は他の作業 a_{3j}, a_{4j} が開始されるまで

Table 1 Data for four activities in Fig. 4

Activity	q_{ij}	TS_{ij}	d_{ij}
a_{1j}	2	1	4
a_{2j}	1	2	4
a_{3j}	3	12	2
a_{4j}	2	13	3

に終了できるため、 $\{a_{1j}, a_{2j}\}$ の順序と $\{a_{3j}, a_{4j}\}$ の順序は独立に考えることができる。したがって、たとえば作業列 $\{a_{1j}, a_{2j}, a_{3j}, a_{4j}\}$ と作業列 $\{a_{1j}, a_{3j}, a_{2j}, a_{4j}\}$ の作業順序は異なっているが、各部分作業群内の順序は同じであるので、これらの作業列は同値関係にあるとみなすことができる。

同値関係にある節点を削除するには、作業群を同値関係にある部分作業群に分割する必要がある。まず作業群をいくつかの独立な部分作業群へ分割する方法について述べる。この部分作業群の中でそれぞれ作業列を生成して統合させることで節点数を減少させることができる。以下に作業群 A を部分作業列 $A^l, l=1, 2, \dots$ へ分割するためのアルゴリズムを示す。

ステップ 1 作業群 A の作業を開始可能時刻の早い順に並べ替え、順に $a(1), a(2), \dots, a(n)$ とおく。

ステップ 2 $l=j=1$ とする。

ステップ 3 作業 $a(i)$ の最遅終了時刻 EF_i を f^* とおき、 $A^l = A^l \cup a(i)$ とする。

ステップ 4 $i=m$ となれば終了する。 $i < m$ のとき、作業 $a(i+1)$ の実施開始可能時刻 TS_{i+1} より f^* が小さければ、 $l=l+1, j=j+1$ としてステップ 3 へもどる。

各部分作業群において同値関係にない作業列では、ある作業の終了時刻が、先行するすべての作業の開始可能時刻より遅くなければならない。この関係を満たさない作業列は削除していけばよい。

3.3 数値モデルによる検証

3.1 と 3.2 の限定操作により分枝限定法を実行することで、各作業群での作業の最適な割り当てを求めることができる。しかし、その計算量は非常に大きく、 m プロジェクトで n 作業の場合では、最大で $(m!)^n$ 通りの組み合わせが考えられる。対象として考察した資源量に時間制約のあるプロジェクト計画問題をランダムに発生させ分枝限定法で解く数値実験を行った。数値モデルは以下のように発生させた。

- (1) ネットワークは各作業の後続作業が決まると一意に定められる。したがって、作業 a_i の後続作業を $a_{i+1}, a_{i+2}, \dots, a_n$ の中から確率 p でランダムに選

ぶことでネットワークを決定した。本実験では $p=0.3$ とした。なお、この操作で後続作業が選ばれなかったときには、 $a_{i+1}, a_{i+2}, \dots, a_n$ から一つをランダムに選んだ。

- (2) 作業時間は $[1, 40]$ の整数値をランダムに与えた。
- (3) 各作業の必要資源量は 1 単位, 2 単位, 3 単位のいずれかをランダムに与えた。
- (4) 資源の使用可能量は、プロジェクトを開始する前ではどの資源に対しても 3 単位とした。

この数値モデルに対して SUN Work Station により最適解を求めようとしたところ、プロジェクト数 m が 3 を越える問題の多くで 120 分の実行時間では最適解を得ることができなかった。

4. 経験則による近似解法

分枝限定法による解法では、小規模の問題しか計画できないことから、経験則にしたがった割り当て優先順位による計画を考える。4. では、資源量に時間制約のあるプロジェクト計画問題に対して、従来より一般のプロジェクト計画問題で用いられているいくつかの経験則にしたがった作業の割り当てに関する数値実験を行い、これらを比較検討した。

4.1 経験則

割り当てアルゴリズムは、第 1 作業群から順に経験則による優先順位にしたがって各作業を実施可能な時間帯へ割り当ててものである。本論文で比較している経験則は以下の八つである。

- 最早終了時刻順 (EFT) ルール
- 最小処理時間順 (SPT) ルール
- 最遅終了時刻順 (LFT) ルール
- 最小スラック順 (SLK) ルール
- 開始可能時刻順 (TS) ルール
- 必要資源量の少ない順 (SNR) ルール
- 必要資源量の多い順 (GNR) ルール
- SNR と GNR を組み合わせた (SGNR) ルール

資源量は考慮せず、使用できる資源が存在しているかないかによって作業実施時間の制約を設けたプロジェクト計画問題では、EFT ルールが有効であることが示されている³⁾。

SPT ルールや SNR ルールは早い時期に多くの作業を実施できるため、後続の作業の開始時間を早めることが期待できる。一方、GNR ルールは資源が不足して開始が遅れる可能性があるものを優先しているため、終了時刻を早めることが期待できる。したがって、GNR ルールはプロジェクトの後半で、SNR ルールはプロジェクトの前半で適用する SGNR ルールについても検証を行った。

4.2 数値シミュレーションの結果

対象とした問題はプロジェクト数を $m=3, 5, 7, 10$ の 4 通り、一つのプロジェクトに含まれる作業数を $n=5, 10, 15, 20, 30, 50$ の 6 通りの全部で 24 通りに設定した。各組み合わせに対して 3.3 で示した方法により五つのネットワークを発生させ、それぞれに 5 種類の作業時間と必要資源量の組み合わせを与えた。全部で 25 個の問題を作り、八つの経験則にしたがったプロジェクト完了時刻の評価を行った。本来なら最適解との比較を行なうべきであるが、ここで扱っている規模の問題に対しては、分枝限定法によって最適解を求めることができなかったため、EFT (最早終了時刻順) ルールによる結果との比によって評価することとした。

Table 2 は、24 通りの設定条件における 25 個の問題に対して各ルールを適用したときの完了時間と EFT ルールによる完了時刻の比の平均値を表したものである。1 より大きな値のルールは EFT ルールより良い結果を与えていることを示している。これから、EFT ルールのほかに LFT (最遅終了時刻順) ルールも比較的良好な結果を示していることがわかる。

資源量を考慮したプロジェクト計画問題においては、LFT ルールが良好な結果を示したのは、開始可能時刻が同じであっても使用可能な資源量が不足して作業の開始が先送りされる場合があるため、最遅終了時刻の方がプロジェクト全体の完了時刻とより強い結びつきがあるものと思われるからである。しかし、作業数が多くなるにつれて、EFT ルールの方が LFT ルールより良い結果を示す傾向がある。これは作業数が多くなると作業時間の短い作業の数も多くなり、これらを空き時間帯で実施できるにもかかわらず最遅終了時刻順に割り当ててために後回しにされる場合が出てくるからと思われる。今回の数値実験では、大規模プロジェクトに対しては EFT ルールが相対的に有効であった。

SNR ルールを前半に適用し GNR ルールを後半に適用する SGNR ルールは、それぞれを単独で適用した場合よりもよい結果を示している。しかし、EFT ルールや LFT ルールより良い解が得られることは少なかった。これはプロジェクトの前半と後半の区別に作業群の番号を用いたことが考えられる。ここで用いた作業群の番号は、作業リスト上の単なる作業番号を表す記号に過ぎず、これによって各作業群をプロジェクトの前半と後半に区別した場合に、それがプロジェクトの実際の進行状況に対応したものになっていないこともあるからである。クリティカルパス上で作業群を前半と後半に分けるなどの工夫をすることで、経験則を組み合わせたルールのよりの確な適用場面を設定することができるであろう。

Table 2 The ratios of completion time; EFT rule over each heuristic rule

m	n	SPT	LFT	SLK	TS	SNR	GNR	SGNR
3	5	0.996	1.043	0.971	0.965	0.926	0.943	0.996
	10	1.010	1.052	0.974	0.990	0.936	0.975	0.990
	15	0.991	1.035	0.985	0.989	0.961	0.971	0.977
	20	0.976	1.014	1.013	0.968	0.949	0.951	0.969
	30	0.995	1.058	0.995	0.978	0.953	0.967	0.993
	50	0.992	1.024	0.997	0.986	0.952	0.960	0.995
5	5	0.993	1.044	0.960	0.945	0.910	0.986	0.995
	10	1.027	0.994	0.963	0.955	0.886	0.924	1.008
	15	0.991	1.039	0.988	0.966	0.940	0.967	0.994
	20	0.984	1.043	0.972	0.964	0.935	0.945	0.979
	30	0.965	1.018	0.974	0.964	0.898	0.913	0.955
	50	0.982	0.994	0.964	0.967	0.902	0.937	0.967
7	5	0.966	1.043	0.953	0.968	0.902	0.917	0.984
	10	0.978	1.026	0.933	0.948	0.913	0.922	0.989
	15	0.972	1.029	0.933	0.977	0.925	0.930	0.984
	20	0.960	1.020	0.955	0.965	0.917	0.952	0.965
	30	0.990	1.017	0.973	0.992	0.929	0.953	0.990
	50	0.965	0.991	0.928	0.976	0.902	0.917	0.957
10	5	1.017	1.022	0.953	0.987	0.924	0.932	1.038
	10	0.988	1.020	0.935	0.949	0.899	0.955	1.012
	15	0.985	1.011	0.943	0.956	0.928	0.933	0.982
	20	0.968	1.007	0.940	0.980	0.922	0.909	0.967
	30	0.946	1.006	0.937	0.954	0.899	0.915	0.933
	50	0.958	0.981	0.927	0.966	0.884	0.912	0.932

5. おわりに

作業を実施できる時間に制約があるプロジェクト計画問題の一つとして、使用可能な資源の量や時間帯に制約のあるときの複数プロジェクト計画問題を考察した。この問題に対する分枝限定法による最適解法では、分枝数を減少させるために同値関係による限定操作を導入し、最適解を導出するアルゴリズムを示した。しかし、並行して計画するプロジェクト数が多くなると組み合わせが莫大な数になり、提案した分枝限定法の効果はまだ確認できていない。今後並列処理などを用いて高速化を図ることが必要となる。

また、経験則による近似解法では、ネットワークの形や資源量制約との関連が明確に結論付けられたのではないが、最遅終了時刻順あるいは最早終了時刻順にしたがって割り当てるルールが良い結果を与えることがわかった。単一のルールで良好な結果が得られるのは限られた場合であるから、より最適解に近づくためには、複数のルールを組み合わせることも必要である。今回の実験で、複

数のルールの組み合わせにより解の改良ができることも確認できた。今後は複数のルールの適用場面のよりの確な設定を検討することも必要である。

参考文献

- 1) C. E. Bell and J. Han: A new heuristic solution method in resource constrained project scheduling; *Naval Research Logistics*, Vol. 38, pp. 315-331 (1991)
- 2) 茨木：アルゴリズムとデータ構造，昭晃堂（1989）
- 3) 金田：確定日程を考慮した複数プロジェクト計画法，神戸大学大学院工学研究科システム工学専攻修士論文（1993）
- 4) I. S. Kurtulus and S. Narula: Multi-project scheduling: analysis of project performance; *IIE Transactions*, Vol. 17, No. 1, pp. 58-66 (1985)
- 5) J. H. Paterson, F. B. Talbot, R. Slowinski and J. Weglarz: Computational experience with a backtracking algorithm for solving a general class of precedence and resource constrained scheduling problems; *European Journal of Operational Research*, Vol. 49, pp. 68-79 (1990)
- 6) F. B. Talbot: Resource constrained project scheduling with time-resource trade offs: The non-preemptive case; *Management Science*, Vol. 28, No. 10, pp. 1197-

1210 (1982)

- 7) J. Weglarz: Project scheduling with continuously divisible double constrained resource; *Management Science*, Vol. 27, No. 9, pp. 1040–1053 (1981)
-