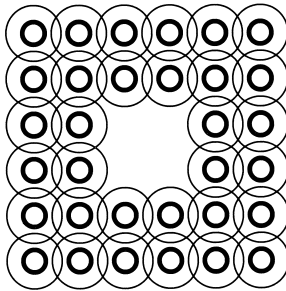


専修経営研究年報



2004年 No. 29

専修大学経営研究所

モデリングシステムを用いた生産計画問題の解法

——モデルを記述しながら整数計画問題を早く解く——

渡 辺 展 男

1. はじめに

トヨタ自動車は2000年7月、トヨタ生産方式における「カンバン方式」として世界的に知られている生産指示方式で特許を取得したことを明らかにした[2]。トヨタ生産方式においては、在庫量の削減、ムダの徹底的排除による原価低減を計ることを基本目標として、「必要な物を、必要な時に、必要な量だけ生産すること」として定義されるジャストインタイム (Just-in-Time; JIT) を実現するために引っぱり型生産指示方式が採用されている[12]。引っぱり型生産指示方式においては、各工程における生産・引き取り指示量はその直後工程で実際に消費された量に基づいて決定されるが、カンバン方式は引っぱり型生産指示方式を実現する一つの情報システムであり、カンバンは生産情報の伝達手段となっている。本稿は、この引っぱり型生産指示方式を対象として、数理計画法によるモデリングについて論じたものである。

数理計画法は、問題解決の科学といわれているオペレーションズ・リサーチ (OR) における代表的な手法の一つである。日本 OR 学会創立 40 周年事業として刊行された「OR 事典 2000」[14] において、近年の 10 年間で発表された OR の応用事例 (約 400 件) が取り上げられ、OR の手法の中で最も用いられている手法はこの数理計画法であること、そして調達・生産・在庫の分野が OR の応用事例の中で最も適用件数が多い分野であることが示されている²⁾。

生産指示方式に関する数理計画法によるモデリングにおいて、その実際問題への適用可能性を検討する場合、次の 2 点、つまり (1) 定式化された問題を解

くための計算量の削減および(2)数式モデルの計算機への入力、モデルとデータの分離等の計算環境の改善が大きな課題となる。対象となる問題は、多くの整数変数を持つ整数計画問題に定式化されるため、前者の解決は重要な課題となる。整数計画問題を解くためのソフトウェアでは、従来その解法として分枝限定法(branch-and-bound method)が採用されていたが、近年では、分枝カット法(branch-and-cut method)と呼ばれる解法が注目されており、商用ソフトウェアにおいても採用されている³⁾。一方、後者の解決方法として1970年代以降開発が進められてきたモデル記述言語といわれるソフトウェアが1990年代以降日本においても入手が可能となったが、近年このモデル記述言語が大きな進展を見せている。その先進性を表すためにモデル記述言語に代わり、新たな進展をみせたソフトウェアに対しては、モデリングシステムという言葉が用いられている⁴⁾。

このような観点から本稿は、著者らが提案した引っ張り型生産指示方式の数理計画モデルを対象として、数値実験を通して近年の数理計画ソフトウェアの求解性能の向上を示すとともに、モデリングシステムの先進性を活用した解法アプローチを提案することをねらいとするものである。本稿の構成は以下の通りである。まず2節でモデリングシステムについて述べた後、3節において整数計画問題として定式化される引っ張り型生産指示方式の数理計画モデルを示す。次に4節においてその整数計画問題を解くために採用している計算手続きを示した後、5節においてモデリングシステムを用いた解法アプローチについて述べる。そして6節においてある自動車部品製造工程を用いて行った数値検証の結果を示した後、最後に7節において本稿のまとめと今後の課題について述べる。

2. モデリングシステム

商用の数理計画ソフトウェアにおける標準の入力データ形式としては、長年MPSフォーマットといわれる特別な形式あるいはこれに準拠した形式が要求されていた。数理計画問題として定式化される数式モデルは制約式の形つまり

行単位で表現される。一方、MPS フォーマットではモデルは変数毎つまり列単位で表現される。従って、実用規模の問題を実際に解こうとする場合、数式モデルから数理計画ソフトウェアへの入力データを生成する過程に関わる計算環境の改善が大きな課題となっていた⁵⁾。この課題に対して、1970年代末以降、数式モデルに近い表現で数理計画ソフトウェアへの入力データとなるマトリックスファイルを生成する試みがなされており、モデル記述言語と呼ばれるソフトウェアが各種開発されてきた。モデル記述言語の利点は、(1)実際に定式化された数式モデルに近い表現でモデルを記述しているため、モデルの修正に対し柔軟に対応できる。(2)モデルとデータがほぼ完全に分離されているため、モデルの構造に変更がない限り入力データを変更するだけで簡単に新たなマトリックスファイルを生成することができ、使用するデータの変更に対し柔軟に対応できる点にある。つまり、モデル記述言語の意義は数理計画ソフトウェアの入力データ形式の標準といえる MPS フォーマットを直接には意識せずにモデルを取り扱うことができるという点にある⁶⁾。

生産指示方式に対する数理計画法によるモデリングにおける課題に対する一つの解決アプローチとして、著者らはメインフレーム上において汎用数理計画ソフトウェアを用い、またワークステーション上においてモデル記述言語を用いて、解法システムを構築し数値計算を通してその有効性を検証した⁷⁾。また2週間を計画期間とする生産計画問題を対象として、パソコン上のシェル環境を用いて構築された解法システムの設計について詳述した[18, 19, 35]。そこではモデル記述言語と最適化モジュールとの連携により、(1)定式化された問題を解くための計算量の削減および(2)数式モデルの計算機への入力、モデルとデータの分離等の計算環境の改善に対する解決が図られている。しかしながら、図1に示すようにその解法システムの枠組みを実現するためには、スクリプト環境など何らかの形で処理全体を制御する仕組みが別途必要となっていた。

近年このモデル記述言語が大きな進展を見せている。その先進性を表すためにモデル記述言語に代わり、新たな進展をみせたソフトウェアに対してはモデ

リングシステムという言葉が用いられている。その特徴を簡潔に表現するとすれば、モデルの記述(Model Describing)とモデルの解法(Model Solving)を一つの環境で実現しているということができる。

Kallrath[28]は、従来のモデル記述言語の役割である最適化モジュールへの入力データとなるマトリクスファイルの生成に加え、モデリングシステムには、例えば求解プロセスを支援する以下のような仕組みが備わっているとしている⁸⁾。

- (1) 最適化モジュールへのコマンド発行による解法の指示
- (2) 実行不可能性の解析
- (3) 解結果のモデルへのフィードバックとその後の手順の指示

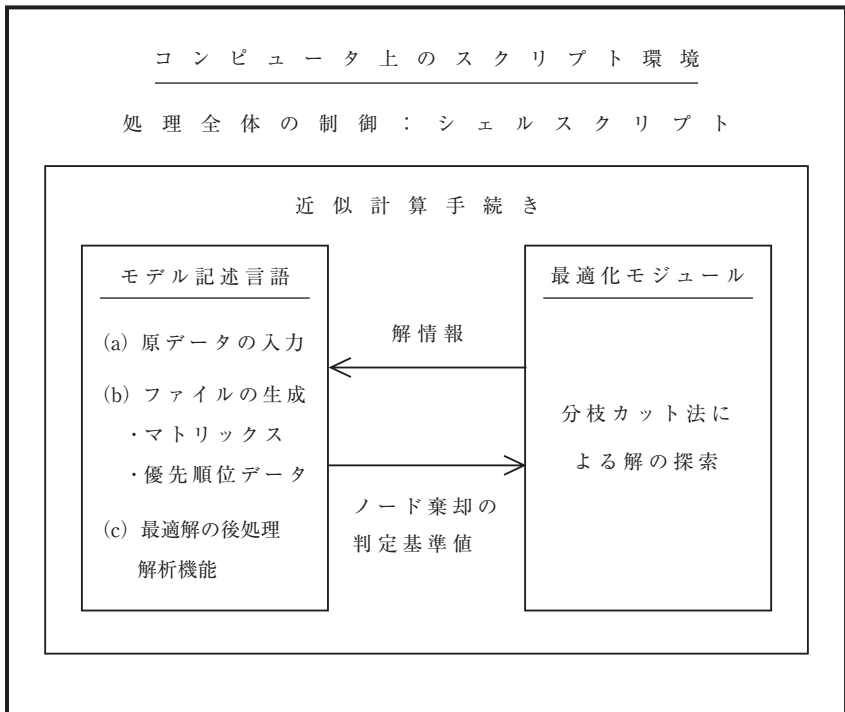


図1 これまでの解法システムの枠組み

(4) 分枝限定 Tree およびマトリックスファイルの表示

以下、本稿では著者らが提案した引っ張り型生産指示方式の数理計画モデルを対象として、代表的なモデリングシステムの一つである Xpress-MP の Mosek[26]を用いた解法アプローチを例示する。

3. 引っ張り型生産指示方式の数理計画モデル

3.1 モデルの条件

本稿で対象とするモデルは、著者らが提案した引っ張り型生産指示方式の数理計画モデルである[17],[37]。このモデルが対象とするシステムは次のような多段階、多品目生産・在庫・運搬システムである。

- (1) 一つの組立工程に収束していく多段工程で N 工程から構成されており、 $n \in \{1, 2, \dots, N\}$ で工程を表す。なお最終工程は $n=1$ とする。
- (2) 各工程は生産工程、加工済み在庫点および後続工程加工待ち在庫点 ($n=1$ では納入待ち製品在庫点) から成る。
- (3) 期間を $t \in \{0, 1, \dots, T\}$ で表す。計画期間は 1 期より始まり T 期で終了する。

またこのモデルは以下の条件で示される生産状況を対象としている。

- (1) 受注先から最終製品の各期の納入量についての内示があり、受注残は認められない。
- (2) 各工程で M 種類の品目が生産される。 $i \in \{1, 2, \dots, M\}$ で品目を表す。
- (3) 各期の各品目について、計画期間全体の生産および引き取り割当量が定まっている。
- (4) 各期の各品目に対する生産および引き取り指示量は前期の期末に計算される。
- (5) 資材在庫は十分にあるが、各工程での生産および引き取りは加工待ちおよび加工済み在庫量の制約を受ける。
- (6) 第 n 工程での生産リードタイムは LP^n である。即ち、 t 期中に生産された品目は $t+LP^n$ 期中に加工済み在庫点に納入される。また引き取り

リードタイムは LH^n である。即ち、 t 期中に引き取られた品目は、 $t+LH^n$ 期中に納入待ち製品在庫点あるいは加工待ち在庫点に納入される。

- (7) 各品目の段取り替え時間および単位量当たり加工時間は既知で計画期間中は一定である。
- (8) 各工程の各品目について期末目標在庫量が設定されている。
- (9) 段取り替えが必要な工程においては、サブロットの大きさが定まっており、生産はこのサブロット単位で行われる。

$N=5$ の場合のモデルの概念図を図 2 に示す。次項 3.2 でモデルの定式化を示すが、この数理計画モデルにおける決定変数は生産および引き取りの初期指示量、つまりカンバン方式における初期投入カンバン枚数であり、その目標は補充目標在庫水準の総和の最小化である。モデルによって決定される初期指示量および補充目標在庫水準のもと、引っ張り型生産指示方式が運用されることとなる⁹⁾。

3.2 定式化

ここで示す数理計画モデルでは、前項 3.1 で述べた記号のほかに次の記号を用いる。

- J : 工程全体の集合 $J = \{1, 2, \dots, N\}$
- $J1$: 最終工程を除いた工程の集合 $J1 = \{2, 3, \dots, N\}$
- K : 段取り替えが必要な工程の集合
- sn : 第 n 工程の直後工程 ($n \in J1$)
- $D_t^{(i)}$: i 製品の t 期の納入内示量
- W_t^n : 第 n 工程の t 期の生産能力(時間)
- $a^n(i)$: 第 n 工程での i 品目の単位量当たり加工時間
- $S^n(i)$: 第 n 工程での i 品目の段取り替え時間 ($n \in K$)
- $L^n(i)$: 段取り替えが必要な工程で加工される i 品目のサブロットの大きさ ($n \in K$)
- $I_0^n(i)$: 第 n 工程での i 品目の初期加工済み在庫量

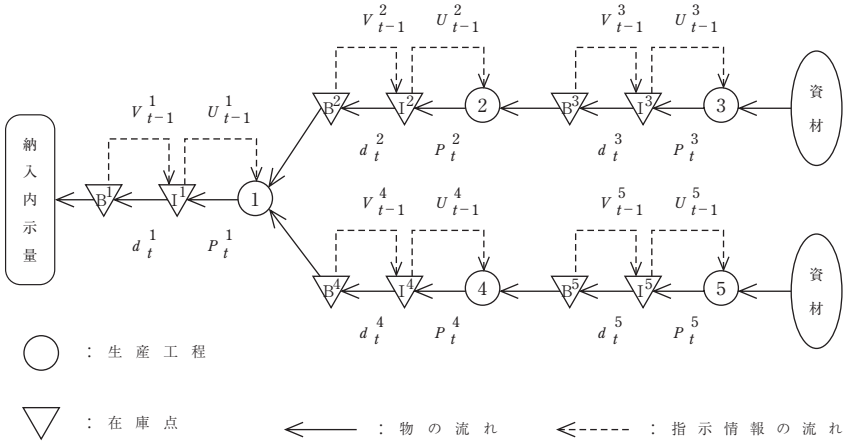


図2 モデルの概念図の例 (N=5)

$B_0^{n(i)}$: i 製品の初期納入待ち在庫量 ($n=1$ の場合) 及び第 n 工程の後工程 sn への i 部品の初期加工待ち在庫量 ($n \in J1$ の場合)

$P_{j-LP^n}^{n(i)}$: 第 n 工程の i 品目についての生産仕掛量 ($j = 1, 2, \dots, LP^n$)

$d_{j-LH^n}^{n(i)}$: 第 n 工程の i 品目についての引き取り仕掛量 ($j = 1, 2, \dots, LH^n$)

$SI_t^{n(i)}$: 第 n 工程の加工済み在庫点における i 品目の t 期末目標在庫量

$SB_t^{n(i)}$: 納入待ち製品在庫点及び加工待ち在庫点における i 品目の t 期末目標在庫量

$Q^n(i)$: 第 n 工程の i 品目についての計画期間全体の生産割当量

$R^n(i)$: 第 n 工程の i 品目についての計画期間全体の引き取り割当量

$e^{sn(i)}$: 直後工程 sn の i 品目を 1 個作るのに必要な第 n 工程の i 品目の個数
 $e^{sn(i)} \in \{1, 2, \dots\}$

なお、納入内示量、在庫量、仕掛量およびサブロットの大きさに関する上記の記号は全て非負の整数である。

$I_t^{n(i)}$: 第 n 工程での i 品目の t 期末における加工済み在庫量

$B_t^{n(i)}$: i 製品の t 期末における納入待ち在庫量 ($n=1$ の場合) 及び第 n 工程の後工程 sn への i 部品の t 期末の加工待ち在庫量 ($n \in J1$ の場合)

- $U_t^n(i)$: 第 n 工程の i 品目について t 期末に計算される $t+1$ 期の生産指示量
 $V_t^n(i)$: 第 n 工程の i 品目について t 期末に計算される $t+1$ 期の加工済み在庫からの引き取り指示量
 $P_t^n(i)$: 第 n 工程での i 品目の t 期中の実際の生産量
 $d_t^n(i)$: 第 n 工程での i 品目の t 期中の実際の引き取り量
 $X_t^n(i)$: 第 n 工程で加工される i 品目についての t 期における段取り替えの回数を表す変数 ($n \in K$)
 $U_0^n(i)$: 第 n 工程の i 品目についての初期生産指示量 (決定変数)
 $V_0^n(i)$: 第 n 工程の i 品目についての加工済み在庫からの初期引き取り指示量 (決定変数)

本稿で対象とする引っぱり型生産指示方式の数理計画モデルは、次のような整数計画問題に定式化される。

Minimize

$$\sum_{n=1}^N \sum_{i=1}^M (I_0^n(i) + \sum_{j=1}^{LP^n} P_{j-LP^n}^n(i) + U_0^n(i) + B_0^n(i) + \sum_{j=1}^{LH^n} d_{j-LH^n}^n(i) + V_0^n(i)) \quad (1)$$

subject to

$$I_t^n(i) = I_{t-1}^n(i) + P_{t-LP^n}^n(i) - d_t^n(i) \quad (i = 1, 2, \dots, M; n \in J; t = 1, 2, \dots, T) \quad (2)$$

$$B_t^1(i) = B_{t-1}^1(i) + d_{t-LH^1}^1(i) - D_t^1(i) \quad (i = 1, 2, \dots, M; t = 1, 2, \dots, T) \quad (3)$$

$$B_t^n(i) = B_{t-1}^n(i) + d_{t-LH^n}^n(i) - e^{sn(i)} P_t^{sn(i)} \quad (i = 1, 2, \dots, M; n \in JI; t = 1, 2, \dots, T) \quad (4)$$

$$U_t^n(i) = U_{t-1}^n(i) - P_t^n(i) + d_t^n(i) \quad (i = 1, 2, \dots, M; n \in J; t = 1, 2, \dots, T) \quad (5)$$

$$V_t^1(i) = V_{t-1}^1(i) - d_t^1(i) + D_t^1(i) \quad (i = 1, 2, \dots, M; t = 1, 2, \dots, T) \quad (6)$$

$$V_t^n(i) = V_{t-1}^n(i) - d_t^n(i) + e^{sn(i)} P_t^{sn(i)} \quad (i = 1, 2, \dots, M; n \in JI; t = 1, 2, \dots, T) \quad (7)$$

$$P_t^n(i) \leq U_{t-1}^n(i) \quad (i = 1, 2, \dots, M; n \in J; t = 1, 2, \dots, T) \quad (8)$$

$$d_t^n(i) \leq V_{t-1}^n(i) \quad (i = 1, 2, \dots, M; n \in J; t = 1, 2, \dots, T) \quad (9)$$

$$P_t^n(i) = L^n(i) X_t^n(i) \quad (i = 1, 2, \dots, M; n \in K; t = 1, 2, \dots, T) \quad (10)$$

$$\sum_{i=1}^M a^{n(i)} P_t^{n(i)} + \sum_{i=1}^M S^{n(i)} X_t^{n(i)} \leq W_t^n \quad (n \in K; t = 1, 2, \dots, T) \quad (11)$$

$$\sum_{i=1}^M a^{n(i)} P_t^{n(i)} \leq W_t^n \quad (n \in J-K; t = 1, 2, \dots, T) \quad (12)$$

$$\sum_{t=1}^T P_t^{n(i)} \geq Q^{n(i)} \quad (i = 1, 2, \dots, M; n \in J) \quad (13)$$

$$\sum_{t=1}^T d_t^{n(i)} \geq R^{n(i)} \quad (i = 1, 2, \dots, M; n \in J) \quad (14)$$

ここで、

$$R^{1(i)} = \max \{0, \sum_{t=1}^T D_t^{(i)} - B_0^{1(i)} + SB_t^{1(i)}\} \quad (i = 1, 2, \dots, M) \quad (15)$$

$$Q^{n(i)} = \max \{0, R^{n(i)} - I_0^{n(i)} + SI_T^{n(i)}\} \quad (i = 1, 2, \dots, M; n \in J) \quad (16)$$

$$R^{n(i)} = \max \{0, e^{sn(i)} Q^{sn(i)} - B_0^{n(i)} + SB_T^{n(i)}\} \quad (i = 1, 2, \dots, M; n \in J1) \quad (17)$$

$$B_t^{1(i)} \geq SB_t^{1(i)} \quad (i = 1, 2, \dots, M; t = 1, 2, \dots, T) \quad (18)$$

$$B_t^{n(i)} \geq SB_t^{n(i)} \quad (i = 1, 2, \dots, M; n \in J1; t = 1, 2, \dots, T) \quad (19)$$

$$I_t^{n(i)} \geq SI_t^{n(i)} \quad (i = 1, 2, \dots, M; n \in J; t = 1, 2, \dots, T) \quad (20)$$

$$X_t^{n(i)} : \text{非負の整数} \quad (i = 1, 2, \dots, M; n \in K; t = 1, 2, \dots, T) \quad (21)$$

$$P_t^{n(i)}, d_t^{n(i)} : \text{非負の整数} \quad (i = 1, 2, \dots, M; n \in J; t = 1, 2, \dots, T) \quad (22)$$

$$U_0^{n(i)}, V_0^{n(i)} : \text{非負の整数} \quad (i = 1, 2, \dots, M; n \in J) \quad (23)$$

評価関数である式(1)で、計画期間中の各工程、各品目の補充目標在庫水準の総和が表され、その最小化を目標としている¹⁰⁾。式(2)～(4)は各在庫点の各期末における在庫量のバランス式である。式(5)～(7)は各工程、各品目の生産指示量、引き取り指示量のバランス式である。またこれらのバランス式(5)～(7)は、各工程における生産・引き取り指示量はその後工程で実際に消費された量に基づいて決定されるという引っ張り型生産指示方式の概念[32]を表現している。式(8)、(9)は指示量による生産量制約および引き取り量制約を示している。式(10)は前項3.1で述べた条件(9)に対応するもので、段取り替えが必要な工程での生産量とサブロットとの関係を表している。式(11)、(12)は生産能力および段取り替え時間による生産量制約である。式(13)、(14)は前項3.1で述べた条件(3)に対応するもので、計画期間全体の割当量による生産・引き取り量制約を表現している。式(15)～(17)はその割当量を定めたものであ

る。式(18)～(20)は各在庫点における期末在庫量に対する制約を表しているが、同時に式(18)は製品納入の保証を表している。また同様に、式(19)、(20)は在庫による実際の生産量と引き取り量に対する制約を意味している。式(21)～(23)は段取り回数、生産量、引き取り量および初期指示量に対する非負整数制約である。

なお $X_t^{n(i)}$, $P_t^{n(i)}$, $d_t^{n(i)}$, $U_0^{n(i)}$, $V_0^{n(i)}$ および納入内示量、初期在庫量、仕掛量およびサブロットの大きさの非負整数性と式(8)、(9)および式(18)～(20)により、各期の指示量 $U_t^{n(i)}$, $V_t^{n(i)}$ および期末在庫量 $I_t^{n(i)}$, $B_t^{n(i)}$ の非負整数性は保証されている。

4. 計算手続き

前節で示したモデルは、在庫量、指示量、生産量、引き取り量および段取り替えの回数に関わる多くの整数変数を持った数理計画モデルである。1節でも述べたように生産指示方式に関する数理計画法によるアプローチにおいて、モデルの構築以降の課題として次の2点があげられる。つまり、(1)定式化された問題を解くための計算量の削減、および(2)数式モデルの計算機への入力、モデルとデータの分離等の計算機環境の改善の問題である。

これら二つの課題を同時に解決する枠組みとして、著者らは2節で述べたようにこれまでモデル記述言語を用いた解法システムを提案してきた。そこでは次の手順を経て問題解決にあたっている。

- (1) 整数変数の数を減らすために、モデルを再定式化する¹¹⁾。
- (2) 再定式化された整数計画問題をモデル記述言語を用いて記述し、最適化モジュールの入力データとなるマトリックスファイルを生成する。
- (3) モデル記述言語と最適化モジュールが提供する機能を統合し、少ない計算量で近似最適解を求めるための近似計算手続きを構築する。

図1で示しているように、その解法システムはモデル記述言語と最適化モジュールとの連携によって実現されているが、コンピュータ上のスクリプト環境など解法を支援するための何らかの仕組みが必要であった。

これに対して、モデリングシステムでは対象とする数理計画問題のモデルとその解き方を同時に記述し問題解決を図ることができる。つまり、モデルの記述 (Model Describing) とモデルの解法 (Model Solving) を一つの環境で実現している。以下、本稿ではモデリングシステム Mosel を用いた新たな解法アプローチを例示するが、その前に本研究で使用している数理計画ソフトウェア Xpress-MP の最適化モジュールである Optimizer [26] (これ以降、Optimizer は最適化モジュールを意味しているものとする) において、初期設定で定められている計算手続きと本研究で採用している近似計算手続きについて述べる。

4.1 標準手続き

前節で示したように、対象とする引っ張り型生産指示方式の数式モデルは整数計画問題となる。一般に整数計画問題は整数変数の数が増えるにつれ、厳密な最適解を得るためには多くの計算量が必要となる。従って、近似最適解を少ない計算量で求めるための何らかの計算手続きが必要となる。

本研究で使用している数理計画ソフトウェアの最適化モジュール Optimizer では、整数計画問題を解くために分枝カット法が採用されている。分枝カット法とは、分枝限定法による探索の過程で切除平面 (cut) を加えながら、緩和問題である線形計画問題を解いていくことで整数解探索の効率化を計ろうとする解法である。いわば分枝限定法と切除平面法 (cutting plane method) の組み合わせと考えられるが、整数計画法の研究においては現在最も注目されているアプローチの一つである¹²⁾。

このように分枝カット法では基本的には分枝限定法の手続きを進行させるため、その計算戦略が計算時間に大きな影響を及ぼす。Optimizer では、パラメータを操作することにより分枝限定法に関わる計算戦略の設定が可能である。

本研究では、次の項目に関するパラメータを操作し計算手続きを作成している。

- (1) ノードの選択
- (2) 分枝変数の選択

(3) ノード棄却の判定基準値

これらの項目に関するパラメータに対して、Optimizer の初期設定は次の通りである。

(1) ノードの選択

- ・下界値優先則と奥行き優先則との折衷則。
- ・つまり、最後に解いたノードの二つの子問題のうち、良いノードを選ぶ。
- ・両方の子問題とも捨てられた場合は、待ちノード全体から良いノードを選ぶ。
- ・良いノードとは、選択の対象となっている子問題の中で最良の下限值(最小化問題の場合)を持つノードとする。

(2) 分枝変数の選択

- ・擬コスト (pseudo-cost) を用いて評価関数の劣化が最も大きいと予想される変数を選ぶ¹³⁾。

(3) ノード棄却の判定基準値

- ・基本的には、最良整数解における評価関数値であるが、式(24)で示される設定であるため評価関数値が同じ整数解は探索されない。

$$\text{CUTOFF} = \text{IPOBJ} + \text{ADDCUT} \quad (24)$$

ここで、CUTOFF：ノード棄却の判定基準値

IPOBJ　：その時点での最良整数解における評価関数値

$$\text{ADDCUT} = \min(-1.0 \text{ E}-5, -1.0 \text{ E}-6 \times \text{LPOBJ})$$

LPOBJ　：変数に対する整数条件が緩和された線形計画問題の最適解における評価関数値。

以下、これらの設定に基づく計算手続きを**標準手続き**と呼ぶ。

4.2 近似計算手続き

標準手続きに対して、近似計算手続きの計算戦略は次の通りである。

(1) ノードの選択

- ・標準手続きと同様、下界値優先則と奥行き優先則との折衷則。

(2) 分枝変数の選択

- ・モデリングシステム Mosel によるモデル記述の中で優先順位を指定する。
- ・その優先順位は順に、(1)段取り替えの回数を表す変数 $X_t^{n(i)}$ 、(2)生産および引き取りの初期指示量 $U_0^{n(i)}$ 、 $V_0^{n(i)}$ 、(3)実際の生産量および引き取り量 $P_t^{n(i)}$ 、 $d_t^{n(i)}$ とする。

なおトヨタ生産方式においては段取り替え作業の取扱いが一つの要点¹⁴⁾となるが、この段取り替えに関わる変数を優先的に分枝させるという優先順位の導入が、本研究で採用している近似計算手続きの特徴¹⁵⁾となっていることが今回行った数値計算においても明らかになっている(後述 6 節を参照)。

(3) ノード棄却の判定基準値

- ・次の式(25)で示される値をノード棄却の判定基準値として設定する。

$$\text{CUTOFF} = \text{IPOBJ} / (1 + \alpha) \quad (25)$$

ここで、CUTOFF：ノード棄却の判定基準値

IPOBJ：その時点での最良整数解における評価関数値

α ：下限値(最小化問題の場合)からの相対誤差。

ここで示した近似計算手続きは、下限値を基準とした相対誤差に基づいた計算手続きである。つまり少ない計算量で相対誤差がある値 α 以内であることを保証する近似最適解が得られる。以下、これらの設定に基づく近似計算手続きを単に**近似手続き**と呼ぶ。切除平面の生成も含めた近似手続きの枠組み(最小化問題の場合)を図 3 に示す。

4.3 数値計算で用いた計算手続き

ここでは、本研究で行った数値計算で用いた計算手続きを示す。各計算手続きの計算結果については 6 節において詳述する。

(1) 標準手続き

4.1 項で述べた標準手続き、即ち Optimizer の初期設定による計算手続き。

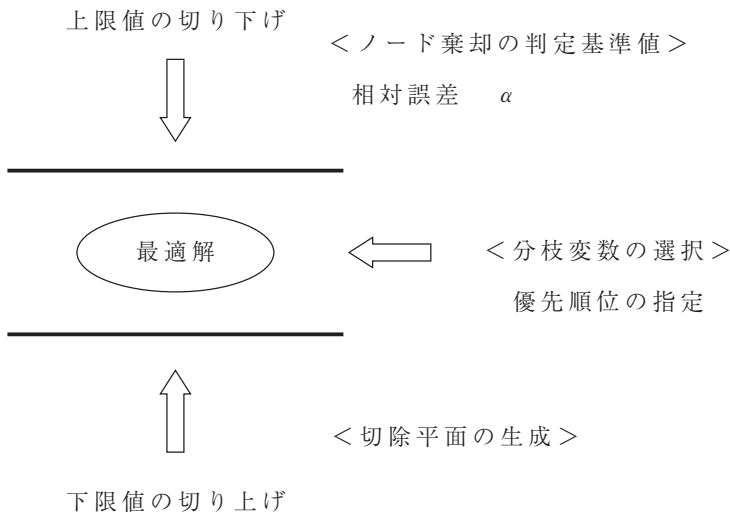


図3 近似手続きの枠組み

(2) 優先順位

分枝変数の選択以外は、標準手続きと同じ計算手続き。またノード棄却の判定基準値の設定以外は、近似手続きと同じ計算手続きでもある。つまりノード棄却の判定基準値については初期設定(4.1項を参照)を用いる計算手続きであり、この計算手続きと標準手続きおよび近似手続きの計算結果を比較することで、近似計算手続きにおける優先順位データの導入の効果そしてノード棄却の判定基準値の設定が計算時間に及ぼす影響が明らかになる。

(3) 近似手続き

4.2項で述べた近似手続きの内、相対誤差 $\alpha=0.01$ とした計算手続き。

5. モデリングシステムを用いた解法アプローチ

モデリングシステムでは図1で示した解法システムの構築をすることなく、対象とするモデルを記述しながら図3で示される近似手続きの枠組みを実現できる。以下では、3.2項で示した引張り型生産指示方式の整数計画問題を対

象として、モデリングシステム Mosel を用いた新たな解法アプローチを例示する。

5.1 モデルを記述する

まず計画期間中の各工程，各品目の補充目標在庫水準の総和を意味する評価関数式(1)は，次のように記述される。なおここでは定数項部分(初期在庫量および仕掛量)を消去している。

! Objective function (1)

$$\begin{aligned} \text{OBJ 1 : =} \\ \text{sum (n in 1..N, i in 1..M) U 0(i, n)} \\ \text{+ sum (n in 1..N, i in 1..M) V 0(i, n)} \end{aligned} \quad (26)$$

次に制約条件の記述であるが，例えば生産能力による生産量制約を表す式(12)は以下のように記述される。なお2行目に示されているのは，工程 n が段取り変え作業の考慮が必要のない工程であれば，この制約を生成しなさいという条件付きの制約生成を表現している。

! Constraints (12)

$$\begin{aligned} \text{forall (n in 1..N, t in 1..T | SK(n) = 0)} \\ \text{C 12(n, t) : = sum (i in 1..M) a(n, i) * P(i, t, n) <= W(n, t)} \end{aligned} \quad (27)$$

上記に示したように，Mosel では数式モデルに近い表現でモデルの記述を行うことができ，これによって Optimizer の入力データであるマトリックスファイルの生成を指示している。

5.2 解き方を記述する

図3で示されている近似手続きのうち，分枝変数の選択に関わる優先順位は次のように記述される。式(28)は段取り替えの回数を表す変数 $X_t^{n(i)}$ に第一優先があることを，式(29)，(30)は生産および引き取りの初期指示量 $U_0^{n(i)}$ ， $V_0^{n(i)}$ に第二優先があることを示している。分枝変数の選択に関わる優先順位を表すパラメータ XPRS_PR の標準値は 500 であり，この値よりも小さな値を

設定することで優先順位を表現できる。なお式(28)の1行目に示されているのは、工程 n が段取り替え作業の考慮が必要な工程の場合、この優先順位の指定が有効であることを示しており、制約の生成のみならずパラメータの設定についても条件付の設定が可能であることを示している。

最後の式(31)はここで設定された分枝変数の選択に関わる優先順位を、あるファイルに書き出すことを示している。この例ではファイル名として `exrel` という名前が設定されている。

```
forall (i in 1..M, t in 1..T, n in 1..N | SK(n) = 1)
    setmipdir(X(i, t, n), XPRS_PR, 100) (28)
```

```
forall (i in 1..M, n in 1..N)
    setmipdir(V0(i, n), XPRS_PR, 200) (29)
```

```
forall (i in 1..M, n in 1..N)
    setmipdir(U0(i, n), XPRS_PR, 200) (30)
```

```
writedirs('exrel') (31)
```

次に図3で示されている近似手続きのうち、ノード棄却の判定基準値の設定については以下のように記述する。これは4.2項で述べたノード棄却の判定基準値の式(25)で示される値を設定するためのもので、手続き名 `setcutoff` としてその手続き (procedure) が記述されている。式(25)に対応させるとパラメータ `XPRS_mipobjval` は整数解が得られた時点での評価関数値 `IPOBJ` を、`XPRS_mipabscutoff` は Optimizer に指示するノード棄却の判定基準値 `CUTOFF` を表している。なお `ALPHA` は相対誤差 α に対応しており、その値はモデル記述の最初の段階で設定しておけばよい。6節で示す数値計算例である近似手続きでは、この値を 0.01 に設定している。

また `getparam` および `setparam` は、Optimizer からその時点でのあるパラメータの値を受け取る (`getparam`) あるいはパラメータを設定し Optimizer へ与える (`setparam`) 役割を果たすものである。Mosel は、このようにパラメータの受け渡しを行うことで Optimizer に対して細かな求解指示を与えることができる。

```

procedure setcutoff
  declarations
    ipobj: real
    cutoff: real
    cutoffnew: real
  end-declarations
  ipobj := getparam('XPRS_mipobjval')
  cutoff := getparam('XPRS_mipabscutoff')
  cutoffnew := ipobj / (1 + ALPHA)
  setparam('XPRS_mipabscutoff', cutoffnew)
end-procedure

```

これらの記述の後、モデリングシステム Mosel は以下のようなコマンドを発行し最適化モジュールである Optimizer に求解の指示を与える。式(33)は callback 機能と言われるもので、整数解が得られた時点で求解を一時停止し、式(32)で示した procedure setcutoff の手続きを実行した後、求解を再開せよという指示を表している。また式(34)の3行目は計算時間(CPU time)が600秒を経過した時点で計算を打ち切れという指示に相当する。最後の式(35)は式(28)–(30)で設定し、式(31)の指示でファイル名 exrel に保存されている分枝変数の選択に関わる優先順位情報を読み取り、評価関数を OBJ 1 とする整数計画問題の最小化(minimize)を実行せよという Optimizer への指示を表している。

```

setcallback(XPRS_CB_INTSOL, 'setcutoff')

```

```

setparam('XPRS_loadnames', true)
setparam('XPRS_verbose', true)

```

```

setparam('XPRS_maxtime', -600)

```

```

loadprob(OBJ 1)
readdirs('exrel')
minimize(OBJ 1)

```

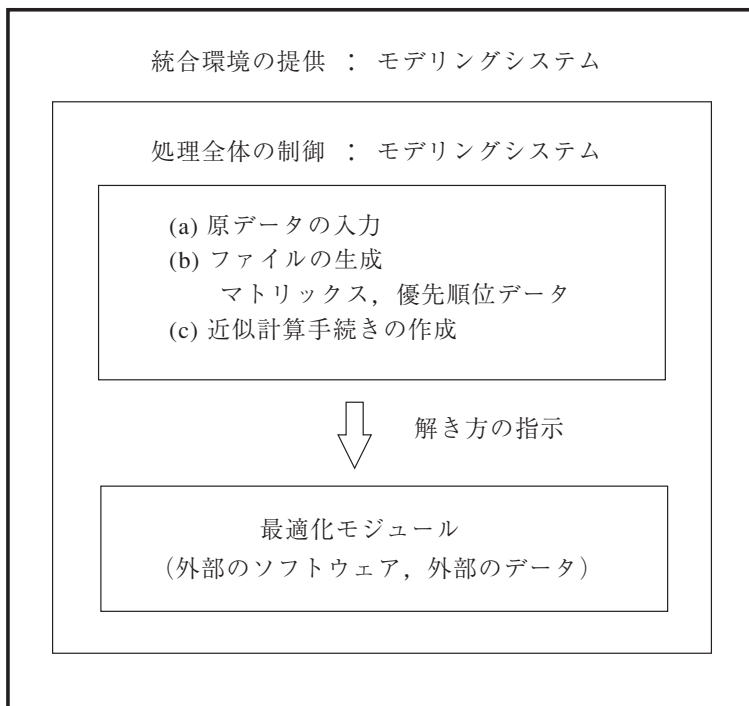


図4 モデリングシステムを用いた解法アプローチ

図4にモデリングシステム Mosel を用いた解法アプローチを示す。これまでのモデル記述言語を用いた場合と異なり、モデリングシステムが処理全体の制御を行っており、モデル記述によるマトリックスファイルの生成のみならず最適化モジュールに対してモデルの解き方を指示する形式となっている。具体的には最適化モジュールとのインターフェースの働きをする `mmxprs` といわれるモジュールを介して最適化モジュールを制御している。上述の `setcallback`, `readdir` および `minimize` などのコマンドも全てそのモジュールを介して `Optimizer` へ伝達されている[26]。

なお Mosel ではインターフェースの働きをする別のモジュールを介在させることで、Xpress-MP の最適化モジュールだけではなく外部のソフトウェアあ

るいは外部のデータを制御することも可能となっている。このインターフェースの働きをする一群の Mosel Modules の存在、つまり modularity の概念がモデリングシステムの機能をさらに拡張させるものとして Mosel の特徴の一つだといわれている [28]。

また図 4 に示しているように、本研究で使用している数理計画ソフトウェア Xpress-MP では、モデルの記述 (Model Describing) とモデルの解法 (Model Solving) を一つの画面上で行う Xpress-IVE といわれる統合環境が提供されている。この統合環境の提供も広い意味でのモデリングシステムの機能といえる。

6. 数値計算例

6.1 数値計算の条件

モデリングシステムの先進性を活用した解法アプローチの有効性を検証するため、ある自動車部品製造メーカーにおける製造工程を対象に数値計算を行った。対象とした製造工程はガソリタンクに使用する小物自動車部品を製造しており次の 5 工程より構成される。その流れ図は図 2 と同じものである。

- (1) 組立工程 (ロー付け) : $n = 1$
- (2) プレス工程 1 (タンデム工程) : $n = 2$
- (3) プレス工程 2 (フープライン) : $n = 3$
- (4) ベンディング工程 (ベンダー) : $n = 4$
- (5) パイプ加工工程 (自動切断機) : $n = 5$.

また、用いた入力データ等の具体的な数値計算の条件は次の通りである。

- (1) 計画期間は 2 週間であり、 t は 1 日単位で $T = 10$ 日である。
- (2) 各工程で生産する品目として、代表的な 3 車種の部品を考える。
- (3) 段取り替えが必要な工程はプレス工程 1 ($n = 2$) およびプレス工程 2 ($n = 3$) である。
- (4) 組立工程 ($n = 1$) およびプレス工程 1 ($n = 2$) における生産リードタイムは 1 日であり、その他の工程における生産リードタイムおよび全ての工程における引き取りリードタイムは十分に短い。

(5) 入力データ

(a) 納入内示量

$$D_t^{(1)} = 20, D_t^{(2)} = 15, D_t^{(3)} = 5 \quad (t=1, 10),$$

$$D_t^{(1)} = 30, D_t^{(2)} = 25, D_t^{(3)} = 5 \quad (t=2, 3, \dots, 9).$$

(b) 生産能力

$$W_t^n = 480 \text{ 分} \quad (n=1, 2, \dots, 5; t=1, 2, \dots, 10).$$

(c) 単位量当たり加工時間

$$a^{n(i)} = 6 \text{ 分} \quad (i=1, 2, 3; n=1, 2),$$

$$a^{n(i)} = 3 \text{ 分} \quad (i=1, 2, 3; n=3, 4, 5).$$

(d) 段取り替え時間

$$S^{2(i)} = 15 \text{ 分}, S^{3(i)} = 10 \text{ 分} \quad (i=1, 2, 3).$$

(e) サブロットの大きさ

$$L^{2(i)} = 10, L^{3(i)} = 10 \quad (i=1, 2, 3).$$

(f) 初期在庫量

$$B_0^{n(1)} = 14, B_0^{n(2)} = 12, B_0^{n(3)} = 5$$

$$I_0^{n(1)} = 14, I_0^{n(2)} = 12, I_0^{n(3)} = 5 \quad (n=1, 2, \dots, 5).$$

(g) 期末目標在庫量

$$SB_0^{n(1)} = 10, SB_t^{n(2)} = 8, SB_t^{n(3)} = 3$$

$$SI_0^{n(1)} = 10, SI_0^{n(2)} = 8, SI_0^{n(3)} = 3 \quad (n=1, 2, \dots, 5; t=1, 2, \dots, 10).$$

(h) 生産仕掛量

$$P_0^{1(1)} = 25, P_0^{1(2)} = 20, P_0^{1(3)} = 5$$

$$P_0^{2(1)} = 30, P_0^{2(2)} = 20, P_0^{2(3)} = 0.$$

なお、部品構成を表す $e^{sn(i)}$ は全て 1 である。

以上の生産条件の下での実際に解くべき整数計画問題の規模は制約式が 656 制約、整数変数は 330 変数となる。

なお、数値計算で用いた計算環境は次の通りである。

- | | |
|---------|--------------------|
| (1) CPU | Pentium M 1.40 GHz |
| (2) RAM | 768 MB |

- (3) オペレーティングシステム (OS) Windows XP Professional SP 1
- (4) 数理計画ソフトウェア Xpress-MP Release 2004 B
- モデリングシステム Xpress Mosel Version 1.4.0
- 最適化モジュール Xpress Optimizer Version 15.20.05
- 統合環境 Xpress-IVE Version 1.15.01

6.2 計算結果と考察

前項 6.1 で示した条件のもとで実施した数値計算の結果を以下に述べる。まず近年の数理計画ソフトウェアの求解性能の向上を示すため、Optimizer の初期設定、つまり標準手続きで実施した計算結果を表 1 に示す。これは昨年の Release 2003 C、Release 2003 G と現時点での Release 2004 B との比較である。表の見方は次の通り。

- (1) CPU Time : 計算手続きの実行に要した計算時間。
- (2) 生成ノード : 分枝限定法の過程で生成されたノードの数。但し、最終結果が計算打ち切りの場合においては、実際の生成ノード数は、その表示数から +100 の範囲にある。
- (3) 評価関数値 : 式(1)の内、定数項(初期在庫量および仕掛量)を除いた評価関数値。

なお、計算時間の中で 0 秒と表示されているのは、1 秒未満であったことを意味している。

表 1 で示されているように、Release 2003 C から Release 2003 G の間で求解性能の進展が著しい。この問題の最適解における評価関数値は 561 である。この初期設定による計算では得られた解が最適解なのかどうかの判定を行うことが出来ず、3600 秒の時点で計算打ち切りにはなっているものの、Release 2003 G ではその最適解を 485 秒の時点で得ている。また Release 2004 B では最適解を得る時点が早まり、1 時間以上の計算時間(3777 秒)を要しているものの最適解に到達しているという判定を行い、計算を完了することが出来ている。著者は 1994 年以来これまで長年この数理計画ソフトウェア Xpress-MP を使用し数

値検証を行ってきたが、標準手続きを用いてこのような求解性能を得たのは初めての経験であった。これまでは、ちょうど表1のRelease 2003 Cの結果のように計算途中で最適解も得られることなく、事前に設定された計算時間の上限による時間制約によって計算打ち切りとなっていた。

本研究で対象としている引っ張り型生産指示方式の整数計画問題の例では、近年の数値計画ソフトウェアの求解性能の進展がこのRelease 2003 GおよびRelease 2004 Bの段階で顕著に現れたわけであるが、現在の分枝カット法の枠組みを基本としたソフトウェアにおける求解性能の向上は、繰り返し解かれる線形計画問題に対する解法の進展と以下に示す技法の積み重ねによるものであるといわれている[3], [22], [24]。

- (1) 切除平面(cut)
- (2) 前処理(preprocessing)
- (3) 分枝変数の選択(variable selection)
- (4) ヒューリスティック(heuristics)

表1 計算結果 (1)

	Rel 2003 C 標準手続き	Rel 2003 G 標準手続き	Rel 2004 B 標準手続き
最初の整数解			
CPU Time	0 秒	4 秒	2 秒
生成ノード	193	138	85
評価関数値	579	634	585
最良の整数解	11 回目	13 回目	10 回目
CPU Time	1223 秒	485 秒	143 秒
生成ノード	592090	68714	15089
評価関数値	566	561	561
最終結果			
CPU Time	3600 秒	3600 秒	3777 秒
生成ノード	1237200	1335100	463164
評価関数値	566	561	561
判定	計算打ち切り	計算打ち切り	最適解判定 探索完了

(5) ノードの選択 (node selection)

本稿では、このうち(3)分枝変数の選択に関わる数値検証を以下に示す。
 表 2 は 4.3 項で示した三つの計算手続きを現時点の Release 2004 B で行った計算結果である(表の見方は表 1 と同じである)。

優先順位の計算手続きは、分枝変数の選択において 4.2 項で示した段取り替えに関わる変数を優先的に分枝させるという優先順位を導入したものである。これによって最適解への到達時間は遅れてはいるものの 1/6 以下の計算時間(603 秒と 3777 秒)で最適解の判定を行い、計算を完了することが出来ている。近似手続きは、さらにノード棄却の判定基準値を導入し少ない計算量で相対誤差がある値 α 以内であることを保証する近似最適解を得ようとする計算手続きである。これにより確かに近似最適解(相対誤差 $\alpha=0.01$)ではあるが、標準手続きに対しては約 1/12(319 秒と 3777 秒)、また優先順位の計算手続きに対しては約 1/2(319 秒と 603 秒)の計算時間で計算を完了することが出来ている。

表 2 計算結果 (2)

	Rel 2004 B 標準手続き	Rel 2004 B 優先順位	Rel 2004 B 近似手続き
最初の整数解			
CPU Time	2 秒	1 秒	1 秒
生成ノード	85	89	89
評価関数値	585	601	601
最良の整数解	10 回目	12 回目	5 回目
CPU Time	143 秒	585 秒	31 秒
生成ノード	15089	73384	3982
評価関数値	561	561	565
最終結果			
CPU Time	3777 秒	603 秒	319 秒
生成ノード	463164	78832	47872
評価関数値	561	561	565
判定	最適解判定 探索完了	最適解判定 探索完了	相対誤差 0.01 探索完了

表1の計算結果より、まず数理計画ソフトウェアの最適化モジュールの求解性能の向上が近年著しいことが示された。また表2の計算結果は、解くべき問題の特徴を踏まえた上で現在考えられている技法を組み入れることで、さらに計算量の削減を図ることが出来ることを示している。本研究が対象としている数理計画モデルはトヨタ生産方式におけるカンバン方式の概念に基づいたモデルであるが、トヨタ生産方式においては段取り替え作業の取扱いが一つの要点となっている。この段取り替えに関わる変数を優先的に分枝させるという優先順位の導入が、計算量を削減させるという結果は極めて興味深いものである¹⁵⁾。

決定変数および補充目標在庫水準

決定変数、つまり計画期間のはじめに提示する第 n 工程の i 品目についての初期生産指示量および初期引き取り指示量は次の通りである。なおこれらの値は、近似手続きにおいて得られている値である。

$$\begin{aligned}
 U_0^{1(1)} &= 31, U_0^{1(2)} = 26, U_0^{1(3)} = 4; & V_0^{1(1)} &= 26, V_0^{1(2)} = 24, V_0^{1(3)} = 3; \\
 U_0^{2(1)} &= 27, U_0^{2(2)} = 29, U_0^{2(3)} = 16; & V_0^{2(1)} &= 26, V_0^{2(2)} = 21, V_0^{2(3)} = 3; \\
 U_0^{3(1)} &= 36, U_0^{3(2)} = 26, U_0^{3(3)} = 10; & V_0^{3(1)} &= 26, V_0^{3(2)} = 26, V_0^{3(3)} = 8; \\
 U_0^{4(1)} &= 26, U_0^{4(2)} = 20, U_0^{4(3)} = 3; & V_0^{4(1)} &= 26, V_0^{4(2)} = 22, V_0^{4(3)} = 3; \\
 U_0^{5(1)} &= 26, U_0^{5(2)} = 20, U_0^{5(3)} = 3; & V_0^{5(1)} &= 26, V_0^{5(2)} = 20, V_0^{5(3)} = 3.
 \end{aligned}$$

この値、初期在庫量および仕掛量によって、提案したモデルの評価関数である各工程における補充目標在庫水準が与えられる。引っ張り型生産指示方式は、この補充目標在庫水準のもとに運用される。

7. まとめ

本稿では、引っ張り型生産指示方式の数理計画モデルを対象として、数値実験を通して近年の数理計画ソフトウェアの求解性能の向上を示すとともに、モデリングシステムの先進性を活用した解法アプローチを提案した。これらは求解の困難な問題に対しては、解くべき問題の特徴を踏まえた上で、複数のアプローチを組み合わせることで対処すれば成果が得られることを例示したものと

なっている。今回作成した近似計算手続きは、いわば(1)分枝カット法によって対象としている問題の解の下限(lower bound)を切り上げる、(2)ノード棄却の判定基準値において α という相対誤差を導入することによって解の上限(upper bound)を切り下げる、さらに(3)解くべき問題の特徴を捉え、段取り替えに関わる変数を優先的に分枝させる優先順位データを導入するという3点の視点から構成されている求解戦略に基づくものである。

著者はモデル記述言語を用いた解法システムの設計に取り組んで以来、数理計画法により現実の問題を実際に解決しようとする場合における以下の観点を繰り返し主張してきた。(1)最適解ではなく近似最適解を出来るだけ短い計算時間で求める。(2)たとえ用いる近似計算法が既存のよく知られた解法を利用したものであっても、解くべき問題の特徴をとらえ解の探索戦略を適切に設定すれば、実用規模の問題を解く道が得られる。(3)定評のあるソフトウェアが提供する個々の機能を統合し、新たな解法システムを設計することで、現実の問題を解くことも合理的なアプローチの一つである。モデリングシステムは数理計画法による問題解決のシステムを設計する上で、新たな解法アプローチを提供してくれている。またそれと同時に最適化モジュールによる適切な求解はもちろん重要であるが、それを実現するためのモデルとデータの入力、計算結果レポートの出力そして問題解決のためのシステム設計が今後より重要であることをあらためて認識させるものとなっている。

本稿では、筆者らが提案した引っ張り型生産指示方式の数理計画モデルを対象として、モデリングシステムを用いた解法アプローチを例示したが、今後はこの方法論を多目標モデルに拡張してみたいと考えている。またモデル記述言語の動向について今後もサーベイを進めていきたいと考えている。

謝辞

本稿は平成15年度専修大学研究助成「数理計画法によるモデリングに関する研究」による研究成果の一部であることを記して、関係各位に厚くお礼申し上げる次第である。

<注>

*本稿中のシステム名および製品名は一般に各社の登録商標または商標です。

- 1) トヨタ生産方式, カンバン方式および引っ張り型生産指示方式については, 秋庭他[1], 平木[4], 黒田他[10], 門田[12], 村松[13]および日本生産管理学会編[15]などを参照するとよい。
- 2) 猪平他[7]の第9章に, 「OR事典2000」で紹介されている応用事例の分類を整理しORの応用分野と用いられる手法について得られた知見が詳しく述べられている。
- 3) 分枝限定法, 分枝カット法については茨木[5], 茨木—福島[6], 今野—鈴木編[8], 久保[9], Beasley(ed.)[23], Carter-Price[25], Martin[31], Nemhauser-Wolsey[33]およびWolsey[39]などを参照するとよい。
- 4) モデル記述言語の近年の進展については, Atamturk-Savelsbergh[22], Kallrath(ed.)[28]などが詳しい。
- 5) MPSフォーマットについては, 前田[11], 反町編[16], 渡辺[17]およびWilliams[38]などを参照するとよい。またLinear Programming FAQのホームページ[30]上にもMPSフォーマットについての解説がある。
- 6) モデル記述言語の詳細については前田[11], 渡辺[17], Kallrath(ed.)[28], Sharda-Rampal[34]およびWilliams[38]などを参照するとよい。またLinear Programming FAQのホームページ[30]上にもモデル記述言語についての解説がある。
- 7) 渡辺[17], 渡辺—安—平木[20], 渡辺—錦織—平木[21], Watanabe-Hiraki[36][37]など。
- 8) Kallrath(ed.)[28]には, 代表的なモデル記述言語の個別の紹介も含め, モデル記述言語の近年の進展とその動向についての詳しい解説がある。
- 9) 決定変数である初期指示量とカンバン方式における初期カンバン枚数との対応については, 渡辺[17], 渡辺—安—平木[20]を参照。
- 10) 評価関数である式(1)で補充目標在庫水準の総和が表されることについては, 渡辺[17], Watanabe-Hiraki[37]を参照。
- 11) モデルの再定式化の詳細については渡辺[17], Watanabe-Hiraki[37]を参照。
- 12) 分枝カット法も含め整数計画法に関する研究の近年の動向については, 藤江[3], Beasley(ed.)[23], Carter-Price[25], Martin[31], Nemhauser-Wolsey[33]およびWolsey[39]などを参照するとよい。また具体的な数値検証も含め整数計画問題に対する求解戦略のサーベイおよびソフトウェアの近年の動向については, Atamturk-Savelsbergh[22], Johnson-Nemhauser-Savelsbergh[27]およびLinderath-Savelsbergh[29]などを参照するとよい。

- 13) 擬コストの詳細については茨木[5], 今野—鈴木編[8]などを参照するとよい。
- 14) 段取り替えの詳細については門田[12]を参照するとよい。
- 15) 本研究で提案している近似計算手続きの特徴については渡辺[17]–[19], Watanabe-Hiraki[36], [37]において詳細な数値検証を加えている。

参考文献

- [1] 秋庭雅夫, 黒田充, 田部勉, 石井和克, 宮崎晴夫, 市村隆哉:「生産管理システムの設計—その研究と活用—」, 日本能率協会, 東京 (1986)。
- [2] 朝日新聞, 2000年7月20日。
- [3] 藤江哲也:“整数計画問題に対する分枝カット法とカットの理論”, オペレーションズ・リサーチ, Vol. 48, No. 12, pp. 935–940(2003)。
- [4] 平木秀作:「自動車の現地生産と部品調達」, 溪水社, 広島 (1996)。
- [5] 茨木俊秀:「組合せ最適化—分枝限定法を中心として」, 産業図書, 東京 (1983)。
- [6] 茨木俊秀, 福島雅夫:「最適化の手法」, 共立出版, 東京 (1993)。
- [7] 猪平進, 齋藤雄志, 高津信三, 出口博章, 渡辺展男, 綿貫理明:「ユビキタス時代の情報管理概論」, 共立出版, 東京 (2003)。
- [8] 今野浩, 鈴木久敏編:「整数計画法と組合せ最適化」, 日科技連, 東京 (1982)。
- [9] 久保幹雄:「組合せ最適化とアルゴリズム」, 共立出版, 東京 (2000)。
- [10] 黒田充, 田部勉, 圓川隆夫, 中根甚一郎:「生産管理」, 朝倉書店, 東京 (1989)。
- [11] 前田英次郎:“数理計画支援システム”, 第5回 RAMP シンポジウム論文集, pp. 57–72 (1993)。
- [12] 門田安弘:「新トヨタシステム」, 講談社, 東京 (1991); Monden, T.: Toyota Production System -An Integrated Approach to Just-In-Time, 3rd ed., Engineering & Management Press (IME), Norcross, Georgia (1998)。
- [13] 村松林太郎:「新版生産管理の基礎」, 国元書房, 東京 (1979)。
- [14] 日本オペレーションズ・リサーチ学会編:「OR事典2000」第2版, 日本オペレーションズ・リサーチ学会 (2001)。
- [15] 日本生産管理学会編:「トヨタ生産方式」, 日刊工業新聞社, 東京 (1996)。
- [16] 反町洋一編:「線形計画法の実際」, 産業図書, 東京 (1992)。
- [17] 渡辺展男:「多段階生産・在庫・運搬システム—数理計画法によるモデリング—」, 溪水社, 広島 (1999)。
- [18] 渡辺展男:“パソコン上のシェル環境を用いた生産計画問題の解法”, 広島大学

- 経済論叢, 第 24 卷, 第 2 号, pp. 53-70(2000)。
- [19] 渡辺展男：“生産計画問題における Cut-and-Branch 法の数値検証”, 広島大学経済論叢, 第 25 卷, 第 1・2 号, pp. 13-29(2001)。
- [20] 渡辺展男, 安范俊, 平木秀作：“引っ張り型生産指示方式の数理計画的アプローチ”, 日本経営工学会誌, Vol. 44, No. 6, pp. 478-486(1994)。
- [21] 渡辺展男, 錦織昭峰, 平木秀作：“モデル記述言語を用いた生産計画問題の解法”, 平成 7 年度第 2 回 OR セミナーテキスト, 数理計画モデルの応用—構築と解法と分析—, pp. 14-28, 日本 OR 学会 (1995)。
- [22] Atamturk, A. and Savelsbergh, M.W.P.: “Integer Programming Software Systems”, to appear in *Annals of Operations Research*.
- [23] Beasley, J.E. (ed.): *Advances in Linear and Integer Programming*, Oxford University Press, Oxford (1996).
- [24] Bixby, R.E.: “Solving Real-World Linear Programs -A Decade and More of Progress”, *Operations Research*, No. 50, No. 1, pp. 3-15 (2002).
- [25] Carter, M.W. and Price, C.C.: *Operations Research -A Practical Introduction*, CRC Press, Florida (2000).
- [26] Dash Optimization : Xpress-MP Getting Started Release 2004 (2004) ;
Xpress-Mosel Language Reference Manual Release 1.4 (2004) ;
Xpress-Optimizer Reference Manual Release 15 (2004) ;
<http://www.dashoptimization.com/>
- [27] Johnson, E.L., Nemhauser, G.L. and Savelsbergh, M.W.P.: “Progress in Linear Programming-Based Algorithms for Integer Programming: A Exposition”, *INFORMS Journal on Computing*, Vol. 12, No. 1, pp. 2-23 (2000).
- [28] Kallrath, J. (ed.): *Modeling Languages in Mathematical Optimization*, Kluwer Academic Publishers, Massachusetts (2004).
- [29] Linderoth, J.T. and Savelsbergh, M.W.P.: “A Computational Study of Search Strategies for Mixed Integer Programming”, *INFORMS Journal on Computing*, Vol. 11, No.2, pp. 173-187 (1999).
- [30] Linear Programming FAQ :
<http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html>
- [31] Martin, R.K.: *Large Scale Linear and Integer Optimization: A Unified Approach*, Kluwer Academic Publishers, Massachusetts (1999).
- [32] Muramatsu, R., Ishii, K. and Takahashi, K.: “Some Ways to Increase Flexibility in Manufacturing Systems”, *International Journal of Production Research*, Vol.

- 23, No.4, pp. 691–703 (1985).
- [33] Nemhauser, G.L. and Wolsey, L.A.: *Integer and Combinatorial Optimization*, John Wiley & Sons, New York (1988).
 - [34] Sharda, R. and Rampal, G.: “Software Survey: Algebraic Modeling Languages on PCs”, *OR/MS Today*, Vol. 22, No. 3, pp. 58–63 (1995).
 - [35] Watanabe, N.: “A PC-based Solution to a Multi-stage Production Ordering System”, *Proc. of the Special International Conference on Production Research* (Special ICPR 2000) provided in a CD-ROM, 6 pages (2000).
 - [36] Watanabe, N. and Hiraki, S.: “A Mathematical Programming Model for a Pull Type Ordering System including Lot Production Processes”, *International Journal of Operations & Production Management*, Vol. 15, No. 9, pp. 44–58 (1995).
 - [37] Watanabe, N. and Hiraki, S.: “A Modeling Approach to a JIT-based Ordering System”, *Annals of Operations Research*, Vol. 69, pp. 379–403 (1997).
 - [38] Williams, H.P.: *Model Building in Mathematical Programming* 4th ed., John Wiley & Sons, Chichester, England (1999).
 - [39] Wolsey, L.A.: *Integer Programming*, John Wiley & Sons, New York (1998).