

チューニング機能を活用した整数計画問題の解法

—引っ張り型生産指示方式の数理計画モデル—

渡 辺 展 男

専修大学経営研究所『専修経営研究年報』第35集 抜刷

平成23年3月発行

チューニング機能を活用した整数計画問題の解法

—引っ張り型生産指示方式の数値計画モデル—

渡 辺 展 男

1. はじめに

本稿は、トヨタ生産方式における「かんばん方式」の概念に基づいた引っ張り型生産指示方式⁽¹⁾の数値計画モデルを対象として、数値計画ソフトウェアを用いた解法、特にチューニング機能を活用した解法について論じたものである。生産指示方式に関する数値計画法によるアプローチにおいて、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、特に前者の解決は重要な課題となる。また後者の課題について、数値計画法を用いて現実の問題を解決しようとする時、それが成功するかどうかは、「解を得る為の計算ではなく、モデル作成者とコンピュータを結び付けるインターフェース如何だ」といわれている[46]。これらの課題に対する一つの解決アプローチとして、著者らはまず計算量の削減を主なねらいとして、メインフレーム上において汎用数値計画ソフトウェアを用いた解法を示した[19][23][44]。次にインターフェースの改善を図るために1970年代以降開発が進められてきたモデル記述言語が、1990年代以降日本でも入手可能となったことに着目し、ワークステーションおよびパソコン上においてモデル記述言語を用いた解法システムの設計・構築を行い、数値計算を通してその有効性を検証した[19]—[21][24][43][45]。さらに、モデル記述言語が2000年代に入り新たな進展をみせモデリン

グシステムという言葉が用いられるようになったことを受けて、モデリングシステムを用いた整数計画問題の解法を例示した[22]。

整数計画問題を解くための数値計画ソフトウェアにおいては、現在、分枝カット法 (branch-and-cut method) が採用されている。分枝カット法とは、整数計画問題の解法として従来から採用されていた分枝限定法 (branch-and-bound method) 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図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⁽²⁾。近年、特に1990年代後半以降、数値計画ソフトウェアにみられる整数計画問題に対する求解性能の向上は、線形計画法、本稿でも述べるノードおよび分枝変数の選択、前処理、切除平面そしてヒューリスティクスなどにおける複数の工夫の積み重ねの成果であるといわれている[2]—[4][27][30]。例えば、代表的な数値計画ソフトウェアである ILOG CPLEX[33]の1988年から2004年までの線形計画法部分における改善率は、アルゴリズムだけで約3360倍といわれている。これに同期のマシンの計算速度の改善率として見込まれる数値1600倍を合わせ考えると、この期間での線形計画法の平均的な改善率は約530万倍であるとの報告がある[2][3][29][30]。

このように数値計画法、特に整数計画法研究の進展にともなう数値計画ソフトウェアの求解性能およびインターフェースの改善は目覚ましいものがあり、現在もその進展は続いている⁽³⁾。また近年では、整数計画問題の求解を支援するためのツールを搭載したソフトウェアもリリースされている。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値計画モデルを対象として、数値計算を通してチューニング機能を活用した整数計画問題の解法を例示するとともに、数値計画法によるモデリングに関する今後の展望についての知見を述べ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値計画モデルを示した後、3節においてこ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す。次に、4節において本研究で使用した数理計画ソフトウェアによる解法およびチューニング機能について述べた後、5節においてある自動車部品製造工程を対象として行った数値検証の結果を示す。そして、最後に6節において本稿のまとめと今後の展望について述べる。

2. 引っ張り型生産指示方式の数理計画モデル

2.1 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[19][45]。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 1 に示す。次項 2.2 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「かんばん方式」における初期投入かんばん枚数であり、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⁽⁴⁾。

2.2 定式化

ここで示す数理計画モデルでは、前項 2.1 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 ($n \in K$)

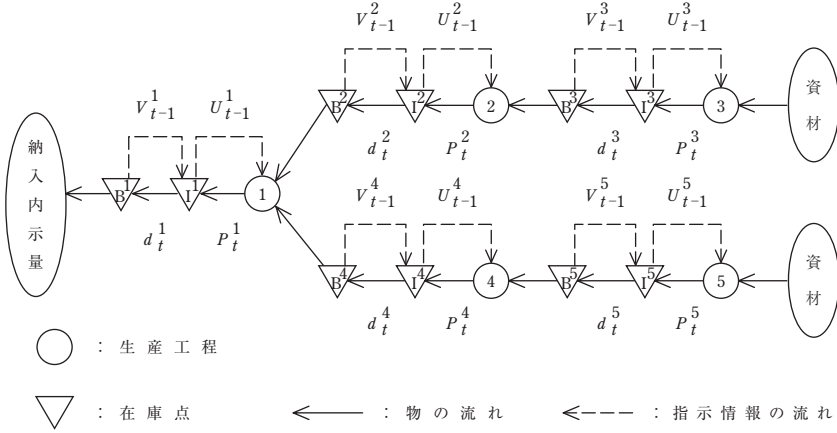


図1 モデルの概念図の例 (N=5)

- $I_0^{n(i)}$: 第 n 工程での i 品目の初期加工済み在庫量
- $B_0^{n(i)}$: 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 ($n \in JI$ 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 ($n \in JI$ 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張り型生産指示方式の数値計画モデルは、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

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

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

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

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

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

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

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

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

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

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

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

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

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

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⁽⁵⁾。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直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張り型生産指示方式の概念[40]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2.1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2.1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)は在庫による実際的な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。

3. 整数計画問題の計算手続き

前節で示した数理計画モデルは, 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整数計画問題に定式化される。従って, 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, 計算量の削減は重要な課題となる。そこで本節では, これまで著者らが例示してきた数理計画ソフトウェアを用いた解法アプローチの変遷を整理する前に, 本研究で現在使用している数理計画ソフトウェア FICO Xpress の最適化モジュールである Xpress-Optimizer[32](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, 整数計画問題を解くために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。

3.1 標準手続き

一般に整数計画問題は整数変数の数が多くなるにつれ, 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値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。1節でも述べたように、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。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⁽⁶⁾。

(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

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。

3.2 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

- ・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 $X_i^{n(i)}$ 、(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 $P_i^{n(i)}$ 、 $d_i^{n(i)}$ 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⁽⁷⁾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⁽⁸⁾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5節を参照)。

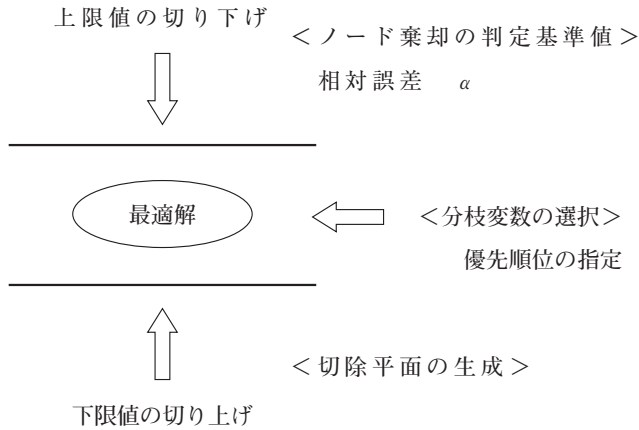


図2 近似手続きの枠組み

(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。

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

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2に示す。

3.3 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5節において詳述する。

(1) 標準手続き

3.1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。

(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(3.1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

3.2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。

4. 数理計画ソフトウェアによる解法

4.1 モデリングシステムを用いた解法

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、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。モデル記述言語の登場によって、上記の二つの課題を同時に解決する枠組みを構築することができるようになったが、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出来る。つまりモデ

リングシステムでは、対象とするモデルを記述しながら、それと同時に図 2 で示される近似手続きを記述することができるのである。

本項では、代表的なモデリングシステムの一つである FICO Xpress の Xpress-Mosel[32] (これ以降、Mosel はモデリングシステムを意味しているものとする) を用いたアプローチを示す。特にモデルの記述を行った後、どのようにノード棄却の判定基準値を設定していくかについて述べる⁽⁹⁾。

図 2 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 3.2 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。次の 5 節で示す数値計算例における近似手続きでは、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 ')

```

(26)

```

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

```

これらの記述の後、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(27)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(26)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(28)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(29)は、モデルの記述部分で Mosel の指示によってファイル名 exdircut.dir 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 OBJ1 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 ') (27)

```

```

setparam(' XPRS_loadnames ', true)
setparam(' XPRS_verbose ', true) (28)
setparam(' XPRS_maxtime ', -600)

```

```

loadprob(OBJ1)
readdir(' exdircut.dir ') (29)
minimize(OBJ1)

```

図3にモデリングシステム Mosel を用いた解法システムの枠組みを示す。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

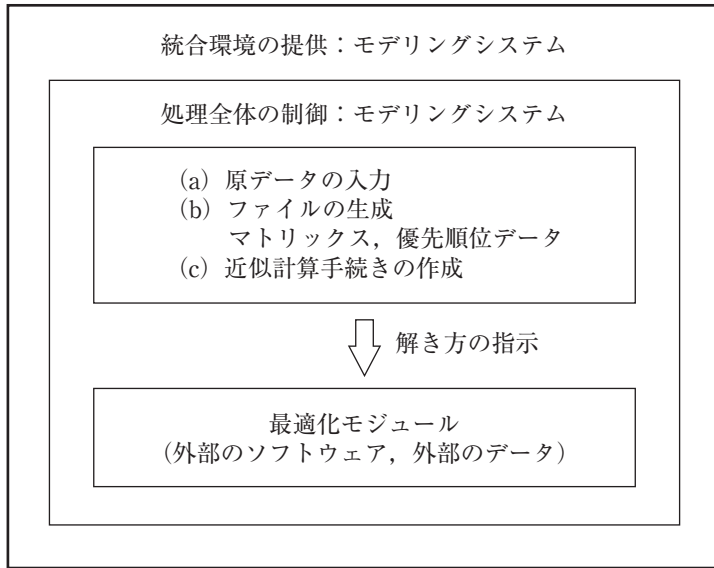


図3 モデリングシステムを用いた解法の枠組み

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。

なお本研究で使用している数理計画ソフトウェア `FICO Xpress` 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。この統合環境も広い意味でのモデリングシステムの機能といえ、数理計画ソフトウェアのインターフェースの改善が進展している証ともいえる。

4.2 チューニング機能

整数計画法研究の進展にともなう数理計画ソフトウェアの求解性能およびイ

インターフェースの改善は目覚ましいものがあり、現在もその進展は続いているが、近年では、整数計画問題の求解を支援するためのツールを搭載したソフトウェアもリリースされている。

数理計画ソフトウェアにおいては、従来から求解に関わる制御パラメータ (control parameter) が多く設けられている。本研究で使用している Optimizer においては、計算結果の出力方法など求解とは直接関係の無いものも含めると、約 160 件の制御パラメータがあるが、FICO Xpress においては Xpress-Tuner[32] (これ以降、Tuner はチューニングツールを意味しているものとする) といわれる整数計画問題の求解を支援するためのツールが用意されている。

分枝カット法の枠組みを基本とした近年の数理計画ソフトウェアにみられる整数計画問題に対す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ともに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[2]—[4][27][30]。

- (1) 切除平面 (cut)
- (2) 前処理 (presolve)
- (3) 分枝変数の選択 (variable selection)
- (4) ヒューリスティクス (heuristics)
- (5) ノードの選択 (node selection)

Tuner の基本的な機能は、対象としている整数計画問題の MPS フォーマットの問題ファイルを用いて、線形計画問題の解法および上記の技法に関わる制御パラメータの設定を変更しながら、ユーザが定めた 1 回の実行時間 (例えば 60 秒)、該当の整数計画問題の計算を繰り返し実行するものである⁽¹⁰⁾。また設定した 1 回の実行時間内に、最適解に到達し計算を終了出来た場合はその計算時間を、計算を終了出来なかった場合は上限値と下限値とのギャップ (gap) が小さいことを、基準としてそれぞれの実行が評価され求解性能の順位付けが行われる。代表的な設定においては、(1)まず、制御パラメータの初期設定 (本稿における標準手続きに相当する設定) による実行を行った後、Phase 1 として 32 種類の制御パラメータの値を 1 件ずつ設定し、計 109 件の設定値に対する

計算を実行する。(2)次に、Phase 2としてPhase 1における計算において、求解性能の優れた上位10位までの制御パラメータのペアの設定で計算を実行する。(3)そして、Phase 3としてユーザがTunerの実行を停止するまでの間、Phase 1からPhase 2の間で得られた制御パラメータの組み合わせを変更しながら、該当の整数計画問題を繰り返し計算していくというものである。数値計算の結果は次節で述べるが、本研究においては、1回の実行時間を60秒と定め、約200回の繰り返し計算を実行し、Phase 3までの計算を試みた。

5. 数値計算例

4節では、モデリングシステムによる解法と本研究で用いたチューニングツールの機能について述べた。1節でも述べたように、近年の数値計画ソフトウェアではインターフェースの改善のみならず整数計画問題に対する求解性能の向上にも著しいものがある。本節では、FICO Xpressの最新リリースを用いて数値計算を行い、最適化モジュールの求解性能およびチューニングツールの機能についての考察を行う。

5.1 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ントank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1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 (バンダー) : $n = 4$
- (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

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

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

(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

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

(e) サブロットの大きさ

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

(f) 初期在庫量

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

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

(g) 期末目標在庫量

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

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

(h) 生産仕掛量

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

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

なお、部品構成を表す $e^{sm(i)}$ 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 | Intel Core i7-620M 2.66 GHz |
| (2) RAM | 4 GB |
| (3) オペレーティングシステム (OS) | Windows 7 Professional 32 bit |
| (4) 数理計画ソフトウェア | FICO Xpress Release 7.0.2 |
| | モデリングシステム Xpress Mosel Version 3.0.3 |
| | 最適化モジュール Xpress Optimizer Version 20.00.21 |
| | 統合環境 Xpress-IVE Version 1.20.12 |
| | チューニングツール Xpress Tuner Version 1.1.0 |

5.2 計算結果と考察

前項 5.1 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、Tuner の実行前において実施した計算結果、つまり 3.3 項で示した三つの計算手続き（標準手続き、優先順位および近似手続き）における計算結果を表 1 に示す。

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。
- (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、最初の整数解とは分枝限定法の手続き以降に最初に得られた整数解という意味である。厳密には、分枝限定法の手続きに入る前の切除平面の導入等で既に整数解は得られる場合もあり、そのために、括弧内に厳密には何回目の

整数解であることも表に示している。

表1で示されているように、Optimizerの最新リリースであるVersion 20.00.21においては、標準手続きにおいても、600秒以内で計算手続きを終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 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る⁽¹¹⁾。

次に、表1の標準手続きの実行の際に生成されたMPSフォーマットの問題ファイルを用いて、Tunerを実行して得られた知見を述べる。標準手続きにおいて、最良の整数解が58秒で得られていることから、Tunerの1回の実行時間を60秒と定め、約200回の繰り返し実行を行った。その結果最も優れた求解が得られる制御パラメータの組み合わせは、次のペアであると判断した。

$$\text{GOMCUTS} = 0 \quad (30)$$

$$\text{HEURDIVESTRATEGY} = 9 \quad (31)$$

表1 計算結果 (1)

	Tuner 実行前 標準手続き	Tuner 実行前 優先順位	Tuner 実行前 近似手続き
最初の整数解	(3回目)	(3回目)	(3回目)
CPU Time	3秒	3秒	3秒
生成ノード	112	86	497
評価関数値	588	613	608
最良の整数解	15回目	20回目	9回目
CPU Time	58秒	20秒	6秒
生成ノード	12672	10378	2290
評価関数値	561	561	565
最終結果			
CPU Time	456秒	197秒	83秒
生成ノード	122347	148183	55463
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

Tuner を用いた 1 回 60 秒の計算内において、全ての実行でいずれも計算を終了出来ていなかったが、最適値 561 に到達し、さらに最良の下限值 559 が得られたのは、約 200 回の繰り返し実行の中で 17 件であった。その中で、式 (30) の設定は全 17 件で現れ、式 (31) の設定は計 8 件において現れた。また、表 1 で示した標準手続きの計算において、下限値が 559 の値に到達するのは 305 秒の時点であり、この二つの制御パラメータのペアを採用することで、求解が早まることが期待される。

なお式 (30) の設定は、ルートノードにおいて Gomory カットを生成しないという意味である。また式 (31) の設定は、ヒューリスティクスの設定、具体的には整数の値になっていない変数を整数値に丸めるための diving heuristic といわれるヒューリスティクスの手続きに関するものであり、最も近い値の整数値に値を切り下げる丸めを、常に行うことを意味している。

Tuner の実行で得られた知見に基づき、式 (30) と式 (31) の制御パラメータをペアで設定して行った計算の結果を表 2 に示す。表の見方は、表 1 と同じである。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。なお式 (30) の設定の影響と見られるが、3 件の計算のいずれも、分枝限定法の手続き以前において整数解は得られていない。

表 1 と表 2 の結果を比べると、チューニングツール活用が有効であることは明らかである。

- (1) 標準手続きにおいては、約 1/5 の計算時間 (12 秒と 58 秒) で最適解に到達し、約 1/5 の計算時間 (95 秒と 456 秒) で計算手続きを終了出来ている。
- (2) 優先順位による計算では、最適解への到達は遅れているものの (47 秒と 20 秒)、約 1/4 の計算時間 (50 秒と 197 秒) で計算手続きを終了出来ている。
- (3) 近似手続きにおいては、約 8 秒の時点で近似解ではなく最適解が得られており、約 1/7 の計算時間 (12 秒と 83 秒) で計算手続きを終了出来ている。

(4) 最終的な比較, つまり, 初期設定に基づく標準手続きと今回チューニングツールを用いて得られた設定に基づく近似手続きの比較においては, 約 1/7 の計算時間 (8 秒と 58 秒) で最適解に到達し, 約 1/38 の計算時間 (12 秒と 456 秒) で計算手続きを終了出来ている。

また, 本研究において主張してきた優先順位による計算および近似手続きの有効性についてもあらためて明らかとなった。優先順位の計算手続きは, 分枝変数の選択において 3.2 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。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「かんぱん方式」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⁽⁸⁾。

表 2 計算結果 (2)

	Tuner 実行後 標準手続き	Tuner 実行後 優先順位	Tuner 実行後 近似手続き
最初の整数解			
CPU Time	1 秒	0 秒	0 秒
生成ノード	131	61	61
評価関数値	635	614	614
最良の整数解	14 回目	20 回目	6 回目
CPU Time	12 秒	47 秒	8 秒
生成ノード	7105	34599	5051
評価関数値	561	561	561
最終結果			
CPU Time	95 秒	50 秒	12 秒
生成ノード	71961	41773	8477
評価関数値	561	561	561
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

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，表 2 の近似手続きの計算において得られている値である。

$$\begin{array}{ll}
 U_0^{1(1)} = 31, U_0^{1(2)} = 29, U_0^{1(3)} = 6; & V_0^{1(1)} = 26, V_0^{1(2)} = 24, V_0^{1(3)} = 3; \\
 U_0^{2(1)} = 27, U_0^{2(2)} = 26, U_0^{2(3)} = 13; & V_0^{2(1)} = 26, V_0^{2(2)} = 21, V_0^{2(3)} = 3; \\
 U_0^{3(1)} = 34, U_0^{3(2)} = 26, U_0^{3(3)} = 10; & V_0^{3(1)} = 26, V_0^{3(2)} = 26, V_0^{3(3)} = 8; \\
 U_0^{4(1)} = 26, U_0^{4(2)} = 20, U_0^{4(3)} = 3; & V_0^{4(1)} = 26, V_0^{4(2)} = 21, V_0^{4(3)} = 3; \\
 U_0^{5(1)} = 26, U_0^{5(2)} = 19, U_0^{5(3)} = 3; & V_0^{5(1)} = 26, V_0^{5(2)} = 20, V_0^{5(3)} = 3.
 \end{array}$$

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。

6. ま と め

本稿では，整数計画問題に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対象として，数値計算を通して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，チューニング機能を活用した整数計画問題の解法を例示した。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，あらためて示すことが出来た。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。そして，チューニング機能を活用して適切な設定をすれば，求解性能がさらに改善され

ることを示した。

数値計算を通して、チューニング機能の活用が有効であることが明らかとなったわけであるが、その一方で、本研究が対象としている整数計画問題は、現在の数理計画ソフトウェアの性能の観点からは、既に求解の面で易しい問題に分類されつつあることがわかる。(1)まず、標準手続きの計算において、600秒以内で計算手続きを終了していること。(2)次に、チューニングツールを用いることで設定した、切除平面の一つである Gomory カットを生成しないことで、求解が早められていることが判明した。他の切除平面に比べ、Gomory カットは有効ではないといわれる場合もあるが、少なくとも本研究が対象とする問題においては切除平面の一部を除去することが求解を早めている。

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。本稿で示したチューニング機能の活用は、数理計画法によるモデリングにおいて、最適化モジュール単体における求解性能の向上はもちろん重要であるが、問題解決に対して適切かつ有効なアプローチを実現するためには、モデルとデータの入力、計算結果レポートの出力などを含め問題解決のための体系的なシステムの提供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。

謝辞

本稿は平成 21 年度専修大学研究助成・個別研究・研究課題「整数計画問題の解法に関する研究」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ます。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, 「かんばん方式」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[5], 黒田他[12], 門田[14], 村松[15], 日本生産管理学会編[16], 大野[17]および大野監修—門田編著[18]などを参照するとよい。
- (2) 整数計画法, 分枝限定法および分枝カット法については, 茨木[6], 茨木—福島[7], 今野[8], 今野—鈴木編[10], 久保[11], Beasley(ed.)[28], Carter-Price[31], Martin[39], Nemhauser-Wolsey[41]およびWolsey[47]などを参照するとよい。また整数計画法の研究における近年の動向については藤江[4], 今野[9], 宮代—松井[13], 柳浦—野々部[26], Johnson-Nemhauser-Savelsbergh[34]およびLinderoth-Savelsbergh[36]などを参照するとよい。
- (3) モデル記述言語を含め数理計画ソフトウェアの近年の進展については, Atamtürk-Savelsbergh[27], Kallrath(ed.)[35], Linderoth-Ralphs[37]およびOR/MS Today誌に掲載されるSoftware Surveys[42]などを参照するとよい。
- (4) 決定変数である初期指示量と「かんばん方式」における初期かんばん枚数との対応については, 渡辺[19], 渡辺—安—平木[23]を参照。
- (5) 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[19], Watanabe-Hiraki[45]を参照。
- (6) 擬コストを含め分枝限定法における計算戦略の詳細については, 茨木[6], 今野—鈴木編[10]などを参照するとよい。
- (7) 段取り替えの詳細については門田[14]を参照するとよい。
- (8) 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[19]—[22], Watanabe-Hiraki[44], [45]において詳細な数値検証を加えている。
- (9) モデルの記述も含めモデリングシステムを用いた解法の詳細については, 渡辺[22]を参照。また, メインフレーム上における汎用数理計画ソフトウェアの場合を含め数理計画ソフトウェアを用いた整数計画問題の解法の変遷については, 渡辺—宇佐美[25]を参照。
- (10) MPSフォーマットについては, 渡辺[19]およびWilliams[46]などを参照するとよい。またLinear Programming FAQのホームページ[38]上にもMPSフォーマットについての解説がある。
- (11) 本稿で用いている計算環境とソフトウェアの設定は異なるわけであるが, 本研究で対象としている数理計画モデルにおける2003年から2007年までの数値計算の結果と考察については, 渡辺—宇佐美[25]を参照。

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] Bixby, R. E.: “ムーアの法則を超えて: かつてない程に短縮された最適化時間”, 講演資料 (2003)
- [3] Bixby, R. E.: “Progress in Optimization & Gurobi Optimizer”, Gurobi Optimizer 販売記念特別講演会資料 (2010)
- [4] 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)。
- [5] 平木秀作:「自動車の現地生産と部品調達」, 溪水社, 広島 (1996)。
- [6] 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)。
- [7] 茨木俊秀, 福島雅夫:「最適化の手法」, 共立出版, 東京 (1993)。
- [8] 今野浩:「整数計画法」, 産業図書, 東京 (1981)。
- [9] 今野浩:「役にたつ一次式—整数計画法「気まぐれ女王」の50年—」, 日本評論社, 東京 (2005)。
- [10] 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)。
- [11] 久保幹雄:「サプライ・チェーン最適化ハンドブック」, 朝倉書店, 東京 (2007)。
- [12] 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)。
- [13] 宮代隆平, 松井知己: “ここまで解ける整数計画”, システム/制御/情報, Vol. 50, No. 9, pp. 363-368 (2006)。
- [14] 門田安弘:「トヨタプロダクションシステム—その理論と体系—」, ダイアモンド社, 東京 (2006)。
- [15] 村松林太郎:「新版生産管理の基礎」, 国元書房, 東京 (1979)。
- [16] 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)。
- [17] 大野耐一:「トヨタ生産方式—脱規模の経営をめざして—」, ダイアモンド社, 東京 (1978)。
- [18] 大野耐一監修, 門田安弘編著:「トヨタ生産方式の新展開」, 日本能率協会, 東京 (1983)。
- [19] 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)。
- [20] 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巻, 第2号, pp. 53-70 (2000)。
- [21] 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)。
- [22] 渡辺展男: “モデリングシステムを用いた生産計画問題—モデルを記述しながら整数計画問題を早く解く—”, 専修経営研究年報, No. 29, pp. 27-55 (2005)。
- [23] 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)。
- [24] 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—構築と解法と分析一, pp. 14-28, 日本 OR 学会 (1995)。
- [25] 渡辺展男, 宇佐美嘉弘: “数理計画ソフトウェアを用いた整数計画問題の解法—引っ張り型生産指示方式の数理計画モデル—”, 専修大学情報科学研究所報, No. 68, pp. 1-25 (2008)。
- [26] 柳浦睦憲, 野々部宏司: “分枝限定法—さらなる計算効率の希求”, システム/制御/情報, Vol. 50, No. 9, pp. 350-356 (2006)。
- [27] Atamtürk, A. and Savelsbergh, M. W. P.: “Integer-Programming Software Systems”, *Annals of Operations Research*, Vol. 140, pp. 67-124 (2005)。
- [28] Beasley, J. E. (ed.): *Advances in Linear and Integer Programming*, Oxford University Press, Oxford (1996)。
- [29] Bixby, R. E.: “Solving Real-World Linear Programs: A Decade and More of Progress”, *Operations Research*, Vol. 50, No. 1, pp. 3-15 (2002)。
- [30] Bixby, R. E., Fenelon, M., Gu, Z., Rothberg, E. and Wunderling, R.: “Mixed-Integer Programming: A Progress Report”, *The Sharpest Cut* (Grötschel, M. ed.), SIAM, Philadelphia, pp. 309-327 (2004)。
- [31] Carter, M. W. and Price, C. C.: *Operations Research: A Practical Introduction*, CRC Press, Boca Raton (2000)。
- [32] FICO: Getting Started with Xpress Release 7 (2010);
Xpress-Mosel Reference Manual Release 3.0 (2009);
Xpress-Optimizer Reference Manual Release 20.00 (2009);
Xpress-Tuner User Guide (2009)。
- [33] ILOG: ILOG CPLEX, <http://www.ilog.co.jp/>; 現在は, IBM ILOG CPLEX。
- [34] 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)。
- [35] Kallrath, J. (ed.): *Modeling Languages in Mathematical Optimization*, Kluwer Academic Publishers, Massachusetts (2004)。
- [36] 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).
- [37] Linderoth, J. T. and Ralphs, T. K.: “Noncommercial Software for Mixed Integer Linear Programming”, *Integer Programming: Theory and Practice* (Karlof, J. K. ed.), CRC Press, Boca Raton, pp. 253–303 (2005).
- [38] Linear Programming FAQ :
<http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html>
- [39] Martin, R. K.: *Large Scale Linear and Integer Optimization: A Unified Approach*, Kluwer Academic Publishers, Massachusetts (1999).
- [40] 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).
- [41] Nemhauser, G. L. and Wolsey, L. A.: *Integer and Combinatorial Optimization*, John Wiley & Sons, New York (1988).
- [42] OR/MS Today Software Surveys: <http://lionhrtpub.com/software-surveys.shtml>
- [43] 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).
- [44] 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).
- [45] Watanabe, N. and Hiraki, S.: “A Modeling Approach to a JIT-based Ordering System”, *Annals of Operations Research*, Vol. 69, pp. 379–403 (1997).
- [46] Williams, H. P.: *Model Building in Mathematical Programming* 4th ed., John Wiley & Sons, Chichester, England (1999).
- [47] Wolsey, L. A.: *Integer Programming*, John Wiley & Sons, New York (1998).