

数理計画ソフトウェアにおけるモデルと 計算手続きの記述について

渡 辺 展 男 (専修大学経営学部)

On Describing of Model and Solution Procedure in Mathematical Programming Software

Norio WATANABE (School of Business Administration, Senshu University)

This paper considers a multi-stage, multi-product production, inventory and transportation system and presents a mathematical programming model of a pull type ordering system based on the concept of kanban system. When we apply a mathematical programming approach to ordering systems, the problem to be solved is twofold. One part is to improve an interface for model solution, as mathematical modeling requires transformations of the model form. The other is to reduce the computational time required, as the mathematical programming model includes many integral variables. The aim of this paper is to discuss the former problem, and specially to illustrate on describing of model and solution procedure in mathematical programming software.

キーワード: 整数計画法, 数理計画ソフトウェア, モデル記述言語, かんばん方式

Key words: Integer Programming, Mathematical Programming Software, Modeling Language, Kanban System

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⁾の数理計画モデルを対象として, 数理計画ソフトウェアにおけるモデルと計算手続きの記述に関する変遷について論じたものである。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, 計算量の削減は重要な課題となる。整数計画問題を解くための数理計画ソフトウェアの多くは, 分枝カット法 (branch-and-cut method) を実装している。分枝カット法は, 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⁾。また数理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 [96]。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・

受付: 2017 年 12 月 11 日

受理: 2017 年 12 月 11 日

構築を行い、数値計算を通してその有効性を検証した [43]-[45], [51], [93], [95]。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 [46]-[48]。

数理計画ソフトウェアは、大別すると、(1) 数理計画問題を解くためのソルバー、(2) モデル作成者とソルバーとの橋渡しの役割を果たすモデル記述言語、そして (3) 各種ツールなどによって構成されている。代表的な数理計画ソフトウェアのソルバーにおいては、近年、CPU のマルチコア環境を活かした並列処理機能が実装されてきており、求解性能のさらなる向上が期待されている。モデル記述言語においては、単にソルバーへ入力データを提供する (モデルの記述) だけの役割に止まらず、ソルバーへの解法を指示 (計算手続きの記述) および開発環境を提供する機能なども合わせもち、2000 年代以降、新たな進展をみせている。さらに整数計画問題の求解における制御パラメータの設定を支援するためのチューニング機能をもったツールなどもリリースされており、数理計画ソフトウェアの機能充実には著しいものがみられ、問題解決のための数理計画法によるアプローチには大きな期待が寄せられている⁽³⁾。

このような観点から、本稿のねらいは、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、数理計画ソフトウェアにおけるモデルと計算手続きの記述に関する変遷を、(1) 1970 年代から 1980 年代において用いられていた汎用数理計画ソフトウェア、(2) 1990 年代以降日本でも入手可能となったモデル記述言語、そして (3) 2000 年代以降新たな進展がみられるモデリングシステムの順に概観するとともに、それらの課題について述べることにある。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理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 節において著者らが使用した汎用数理計画ソフトウェア、モデル記述言語およびモデリングシステムにおけるモデルと計算手続きの記述について述べる。そして最後に 5 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

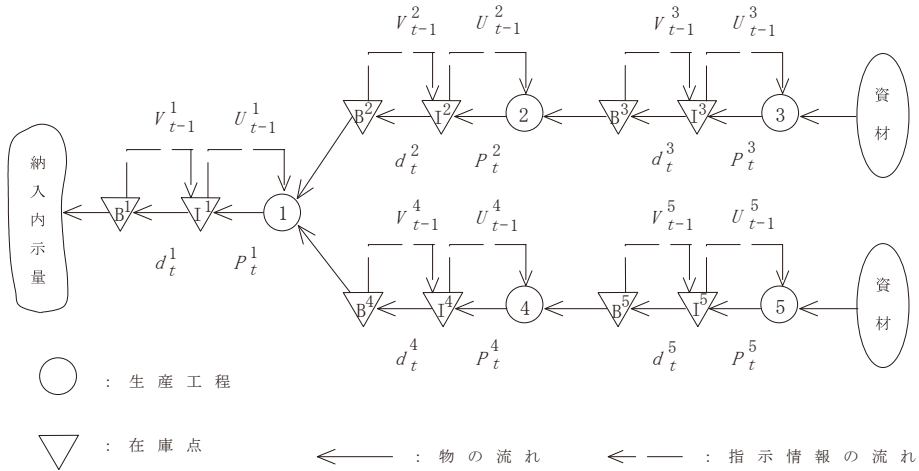
2.1 モデルの条件

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 [43], [95]。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。

図1 モデルの概念図の例 ($N=5$)

- (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、この数値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

ここで示す数値計画モデルでは、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 ($n \in K$)

$I_0^{n(i)}$: 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) および第 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およ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, 納入内销量, 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) および第 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

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

subject to

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

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

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

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

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

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

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

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

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

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

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

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

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

ここで,

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

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

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

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

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

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

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

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

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

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 [87] 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。

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、本研究で従来から使用している数理計画ソフトウェア FICO Xpress のソルバーである Xpress-Optimizer [68]（これ以降、単に Optimizer と記述した場合は、Xpress-Optimizer 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理計画ソフトウェアのソルバー 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、制御パラメータ（control parameter）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、次の項目に関する制御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制御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、奥行き優先則そして複数の下界値優先則と奥行き優先則との折衷則が用意されているが、実際にどの選択ルールを採用するかについては、入力されたデータ構造の特性によって、Optimizer が自動的に決定する⁽⁶⁾。
- (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⁽⁷⁾。
- (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⁽⁸⁾。

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

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。

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。
- (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 $X_t^{n(i)}$, (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⁽⁹⁾ 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となっていることがこれまで行った数値計算においても明らかになっている⁽¹⁰⁾。
- (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
(例えば $\alpha = 0.01$)。

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。

次節では、著者らがこれまで使用してきた数理計画ソフトウェアにおけるモデルと計算手続きの記述に関する変遷を、(1) 1970 年代から 1980 年代において用いられていた汎用数理計画ソフトウェア、(2) 1990 年代以降日本でも入手可能となったモデル記述言語、そして (3) 2000 年代以降新たな進展がみられるモデリングシステムの順に述べる。なお計算手続きの記述については、本項の近似手続きの実現方法という観点から述べることとする。

4. モデルと計算手続きの記述

4.1 汎用数理計画ソフトウェアにおける記述

本項では、主に 1970 年代から 1980 年代において用いられていた汎用数理計画ソフトウェア MPS (これ以降、MPS という言葉は汎用数理計画ソフトウェアを意味しているものとする) におけるモデルお

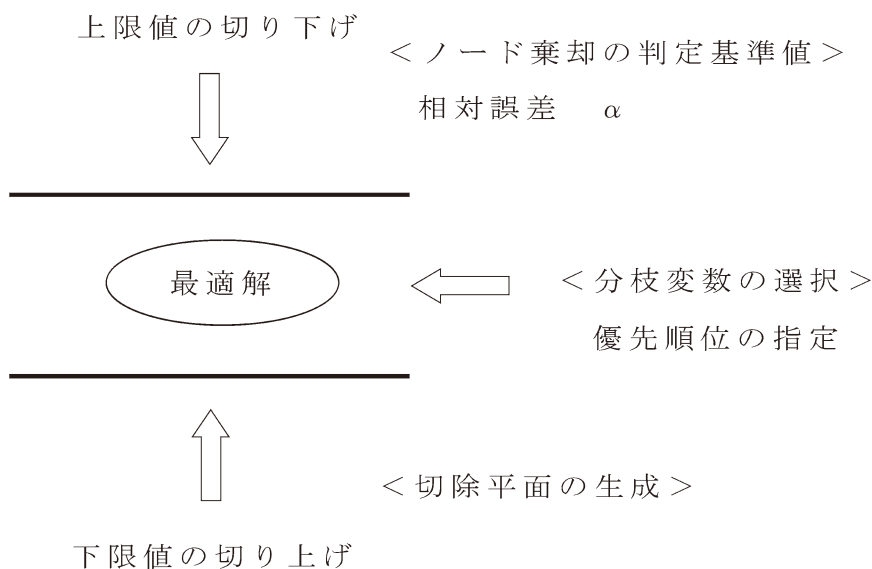


図2 近似手続きの枠組み

よび計算手続きの記述について述べる⁽¹¹⁾。

汎用数値計画ソフトウェア MPS においては、一般には対象とするモデルを記述する環境は用意されておらず、入力データとして MPS フォーマットといわれる業界標準の形式が要求されていた⁽¹²⁾。ここでは以下に示す生産計画問題を用いて MPS フォーマットの簡単な例を示す [43]。

<生産計画問題>

ある工場では3種類の製品 P_1 , P_2 , P_3 を生産しており、おのこの製品を製造するために2種類の原料 M_1 , M_2 が必要である。これらの製品に対する生産計画を作成するにあたり、次のようなことがわかっている。

- (1) 製品 P_1 を 1 kg 生産するためには、原料 M_1 が 1 kg, M_2 が 1 kg 必要である。
製品 P_2 を 1 kg 生産するためには、原料 M_1 が 1 kg, M_2 が 3 kg 必要である。
製品 P_3 を 1 kg 生産するためには、原料 M_1 が 4 kg, M_2 が 5 kg 必要である。
- (2) 工場の倉庫に在庫として、原料 M_1 を 9 kg, M_2 を 10 kg 持っている。
- (3) 製品 P_1 を 1 kg 生産すれば 1 万円、製品 P_2 を 1 kg 生産すれば 2 万円、製品 P_3 を 1 k 生産すれば 3 万円の利益が見込まれる。

このような条件のもとで利益を最大にするには、製品 P_1 , P_2 , P_3 をどれだけ製造すればよいかを決定したい。

製品 P_1 , P_2 , P_3 の生産量をそれぞれ x , y , z kg として、この生産計画問題を定式化すると、次のようになる。

$$\text{最大化} \quad x + 2y + 3z \quad (26)$$

$$\text{制約条件} \quad x + y + 4z \leq 9 \quad (27)$$

$$x + 3y + 5z \leq 10 \quad (28)$$

$$x, y, z \geq 0 \quad (29)$$

上述の<生産計画問題>では、MPS フォーマットのデータ形式は例えば図3に示す形式のものとな


```

1      5      15      25      40      50
(桁)

NAME      PLAN
ROWS
N  PROFIT
L  CAP1
L  CAP2
COLUMNS
      X      PROFIT      1.0      CAP1      1.0
      X      CAP2      1.0
      Y      PROFIT      2.0      CAP1      1.0
      Y      CAP2      3.0
      Z      PROFIT      3.0      CAP1      4.0
      Z      CAP2      5.0
RHS
      RHS      CAP1      9.0      CAP2      10.0
ENDATA

```

図3 MPS フォーマットの例

る。なお (1) PROFIT, CAP1, CAP2 は式 (26)～(28) に対応する行名, (2) X, Y, Z は変数 x , y , z に対応する列名, そして (3) RHS は式 (27), (28) の右辺ベクトルに対応する右辺名を表している。MPS フォーマットは業界標準のデータ形式であるため (現在でも標準形式として使うことができる) 数理計画ソフトウェア間のデータ互換性をほぼ保証するものである。しかしながら, モデルとデータが分離されず, 変数毎つまり列単位でそのまま記載されたデータ形式であるため, モデル作成者は入力データの作成に困難をきたすこととなる。使用する数理計画ソフトウェアに, MPS フォーマットの入力データを生成する何らかの機能がない場合, 例えば著者らが過去に経験した整数変数 210, 制約条件 355 程度の整数計画問題の数値実験では, 約 2,200 行の MPS フォーマットの入力データファイルを手作業で作成していた。また用いる原データが変更となった場合, その都度該当箇所を探索し変更する作業が必要となる。

一方, 次項で述べるモデル記述言語を用いた場合, 定式化した数式モデルに近い表現の記述をするのみで短時間の内に, 使用する数理計画ソフトウェアが要求する MPS フォーマット準拠の入力データを作成することができる。しかも原データが変更となった場合, 基本的には作成したモデル記述は何ら変更することなく, 該当の原データを変更するのみで新たな入力データの作成を行うことができ, 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の改善が図られている。

次に, 汎用数理計画ソフトウェア MPS において 3.2 項で述べた近似手続きを実現する計算手続きの記述について述べる。汎用数理計画ソフトウェア MPS を用いて近似手続きを実現するための解法システムの枠組みは図 4 に示す通りである。処理全体の制御は, 計算機システムのオペレーティングシステム (OS ; Operating System) でのジョブ制御言語 (JCL ; Job Control Language) の記述により行われ, 近似手続きの制御は MPS での手続き制御言語の記述に従い行われる⁽¹³⁾。図 4 において入力データファイルの内, マトリックスとあるのは定式化された整数計画問題における係数マトリックス等に対応するデータファイルを意味しており, 優先順位データとあるのは近似手続きで用いた分枝変数の選択のための優先順位に対応するデータファイルのことである。既に本項で述べたように汎用数理計画ソフ

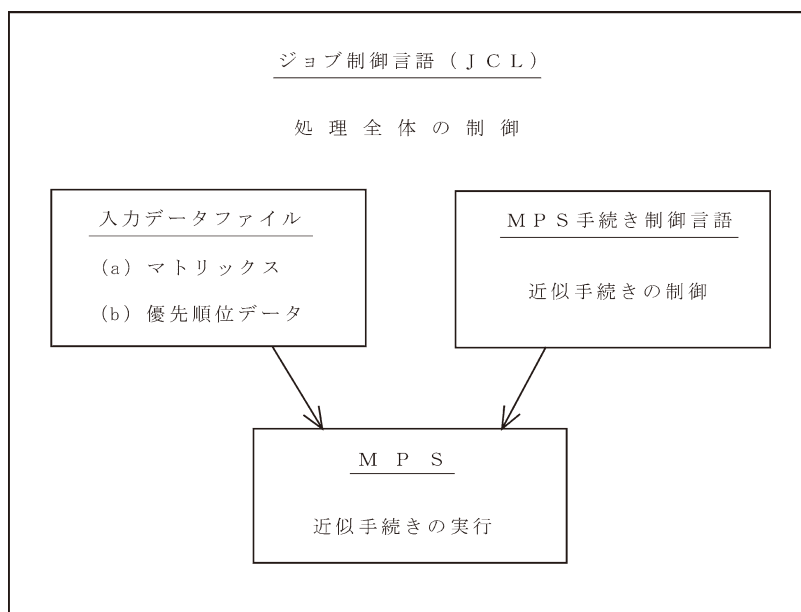


図4 MPSにおける解法システムの枠組み

トウェア MPS では、このデータファイルの形式として MPS フォーマットのデータ形式が要求される。

実際に著者らが使用した汎用数理計画ソフトウェア MPS [32] における MPS 手続き制御言語による近似手続きの記述を図5に示す。なお図5の見方は記述順に示すと次の通り。

(1) PROGRAM

MPS 手続き制御言語による記述の始まりを示す。

(2) TITLE

標準出力ファイルの各ページの先頭に書き出される見出しを指定する。

図5の例では、*** BASIC MODEL *** と指定されている。

(3) CALL ATTACH

ATTACH 手続きは使用するファイルの定義を行う。図5での例は以下の通り。

・内部ファイル名：MP01

入力データファイルを内部処理用の形式に変換した問題ファイルを定義する。

・内部ファイル名：MP02

係数マトリックス等に対応する入力データファイルを定義する。

(4) ADATA, APBNAME

解く問題および問題データの名前を指定する。図5の例では BASIC という名前が指定されている。

(5) CALL CONVERT

CONVERT 手続きは内部ファイル名 MP02 のファイルから入力データを受け取り、内部処理用の形式に変換し、そのファイルに指定された名前 (BASIC) を付けて、問題ファイルとして登録する。

(6) CALL SETUP

SETUP 手続きは計算および計算結果の出力に必要な初期設定を行う。

```

PROGRAM
TITLE   *** BASIC MODEL ***
CALL ATTACH(FILE='MP01',PROFILE,NEW)
CALL ATTACH(FILE='MP02',SOURCE,OLD)
ADATA='BASIC'
APBNAME=ADATA
CALL CONVERT(FILE='MP02',SOURCE)
CALL SETUP(MIN,RANGE='RNG',BOUND='BND')
AOBJ='OBJ1'
ARHS='RHS'
CALL PRIMAL
CALL SOLUTION
CALL RESET(FREQ,TOLERN,INTRUPT,SWITCH)
IMXSW3=0
GO TO 14
3  ITIME=0
   CALL SOLUTION
   FMXDROP=FMXBIF / (1 + 0.01)           * ノード棄却の判定基準値 *
   RETURN
13 ASAVNAM='SAVE****'
   CALL SAVE
   CALL MIXSTATS
   STOP
14 ASSIGN 3 TO KSOLTN                    * 割り込み処理 (文番号 3 へ) *
   ASSIGN 13 TO KMXOVFL
   ASSIGN 13 TO KTIME
   CALL ATTACH(FILE='MP03',SOURCE,OLD)
   ATADA='BASIC'
   CALL MIXSTART(ORDER,SOURCE,FILE='MP03') * 優先順位の指定 *
   ILOGP=0
   ITIME=300
   CALL MIXFLOW
   ILOGP=1
   CALL MIXSTATS
   STOP
END

```

図 5 MPS 手続き制御言語による近似手続きの記述例

なお図 5 の例ではパラメータで最小化問題であること (MIN)、制約式に上下制限があること (RANGE) および変数に上下制限制約があること (BOUND) が指定されている。

- (7) APBJ, ARHS
評価関数の行名, 右辺ベクトルの列名を指定する。
- (8) CALL PRIMAL
PRIMAL 手続きは改訂シンプレックス法の計算手続きを実行し最適解を求める。
- (9) CALL SOLUTION
SOLUTION 手続きは計算により得られた解を出力する。
- (10) CALL RESET
RESET 手続きは MPS の計算手続きの制御パラメータである CR 変数の値を, 初期値に戻す。

(11) IMXSW3

分枝限定法の計算手続きにおける探索方法の設定に関わる CR 変数の一つ。

なお図 5 の例では値 0 を指定し、「ノード選択については、1 回目の整数解到達までは奥行き優先則、それ以降は下界値優先則と奥行き優先則の折衷規則」という設定を行っている。

(12) 文番号 3 の処理

分枝限定法により整数解が得られ、解の出力要求があった時の手続きを指定している。

・ ITIME

時間制限による割り込み処理を起こすまでの時間を秒単位で指定する CR 変数。図 5 の例では値 0 を指定し、整数解が一つでも求まれば、これ以降は時間制限による何らかの割り込み処理は発生させないことを指定している。

・ CALL SOLUTION

現在の解を出力する。

・ FMXDROP

分枝限定法の計算手続きにおける探索方法の設定に関わる CR 変数の一つ。

図 5 の例では、「ノード棄却の判定基準値については得られた最良整数解（暫定解）の評価関数値よりやや良い値、つまり近似最適整数解を求める」という近似手続きの設定を行っている（3.2 項、特に式 (25) を参照）。

(13) 文番号 13 の処理

分枝限定法の計算手続きの過程で生成されるノード情報を蓄積するファイルがあふれた時あるいは時間制限により、割り込み処理が発生した時の手続きを指定している。

・ ASAVNAM

MPS のリスタート情報の名前を指定する。

・ CALL SAVE

SAVE 手続きは MPS のリスタート情報を保存する。

・ CALL MIXSTATS

MIXSTATS 手続きは分枝限定法による整数解探索の現在の状態を出力する。

(14) 文番号 14 の処理

・ KSOLTN

解を出力する時に発生する割り込みに関わる CR 変数。

分枝限定法の計算手続き（具体的には、後述の MIXFLOW 手続き）では、整数解が得られる毎に解を出力させるための割り込みが発生し、MPS の手続きが KSOLTN で指定された飛び先へ移動する。図 5 の例では、文番号 3 が指定されている。

・ KMXOVFL

分枝限定法の計算手続きの過程で生成されるノード情報を蓄積するファイルがあふれた時に発生する割り込みに関わる CR 変数。

MPS の手続きは KMXOVFL で指定された飛び先へ移動する。図 5 の例では、文番号 13 が指定されている。

・ KTIME

時間制限による割り込みに関わる CR 変数。CR 変数 ITIME で指定された時間が経過した時の MPS の手続きの飛び先が指定されている。図 5 の例では、文番号 13 が指定されている。

(15) CALL ATTACH

ATTACH 手続きは使用するファイルの定義を行う。ここでの例は以下の通り。

- ・ 内部ファイル名：MP03

分枝変数の選択のための優先順位データに対応する入力データファイルを定義する。

(16) CALL MIXSTART

MIXSTART 手続きは分枝限定法で整数解を求めるのに必要な初期設定を行う。なお、図 5 の例ではパラメータ (ORDER) で「分枝変数の選択については優先順位データ順」という近似手続きの設定を行っている (3.2 項参照)。

(17) ILOGP

反復情報の出力頻度を表す CR 変数。値 0 は反復情報を出力しない、値 1 は反復毎に出力を行うことを意味している。

(18) ITIME

(12) で述べた通り、時間制限による割り込み処理を起こすまでの時間を秒単位で指定する CR 変数。この例では値 300 とし、300 秒経過しても整数解が一つも得られない場合、割り込みを発生させ CR 変数 KTIME が指定している飛び先へ手続きを移動させる措置を取っている。なお (12) で述べた通り、整数解が一つでも得られれば、この値は 0 に指定され時間制限による割り込み処理は発生させない措置を取っている。

(19) CALL MIXFLOW

MIXFLOW 手続きが分枝限定法の計算手続きを実行する。

(20) STOP

MPS 手続きの実行を終了する。

(21) END

MPS 手続き制御言語による記述の終わりを示す。

汎用数値計画ソフトウェア MPS においては、入力データとして MPS フォーマットといわれる業界標準の形式が要求されるため、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環境は未整備の状況である。その一方で計算手続きの記述については、図 5 で示されているように MPS 手続き制御言語によって比較的簡潔に近似手続きが実現しているといえよう。この点において興味深いのは、4.3 項で述べるモデリングシステムにおける近似手続きの実現方法と似通った計算手続きの記述によって解法システムが構成されているということである。

4.2 モデル記述言語における記述

前項で述べたように汎用数値計画ソフトウェアでは、一般に入力データ形式として MPS フォーマットに準拠した特別な形式が要求される。2 節で示した引っ張り型生産指示方式の数値計画モデルのように、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 フォーマットにおいては、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、数式モデルから数値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る。

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値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いる。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⁽¹⁴⁾。以下、モデル記述言語 mp-model（本研究で使用した数理計画ソフトウェアのモジュール名であるが、これ以降 mp-model はモデル記述言語を意味しているものとする）によるモデルの記述および近似手続きの実現方法について述べる。

モデルを記述する

モデル記述言語 mp-model では、まず TABLES セクションで問題の入力データとなるパラメータ部分の記述を行った後、DISKDATA セクションで外部ファイルを読み込み、TABLES セクションで定義したテーブルにデータを格納する。例えば以下のような記述を行う。

TABLES

SK(N)	! Set of indices representing whether or not ! the production process of the stage ! requires setup !!! SK(n) = 1, if stage n requires setup !!! SK(n) = 0, if stage n does not require setup
SN(N)	! Set of the immediately succeeding stage
DM(T, M)	! Demand for final products
W(N, T)	! Production capacity
a(N, M)	! Processing time per unit
S(N, M)	! Setup time
L(N, M)	! Size of subplot
I0(N, M)	! Initial inventory quantity ! at succeeding inventory points
B0(N, M)	! Initial inventory quantity ! at on-hand inventory points
SI(N, T, M)	! Inventory level at the end of each period
SB(N, T, M)	! On-hand inventory level at the end of each period
e(N, M)	! Bill of materials

DISKDATA

SK = ask.dat	! Files for input data
SN = asn.dat	
DM = adm.dat	
W = aw.dat	


```

a = aa.dat
S = as.dat
L = al.dat
I0 = ai0.dat
B0 = ab0.dat
SI = asi.dat
SB = asb.dat
e = ae.dat

```

次に CONSTRAINTS セクションで、評価関数と制約条件の記述を行う。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。また式 (30) の最後の記号 \$ は、この式が目的関数であることを示している。

CONSTRAINTS

! Objective function (1)

$$\text{OBJ1 : } \sum(n = 1 : N, i = 1 : M) U0(i, n) + \sum(n = 1 : N, i = 1 : M) V0(i, n) \$ \quad (30)$$

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。なお記号 & は、以下に記述が継続することを示している。

! Constraints (12)

$$\begin{aligned} & \text{C12}(n = 1 : N, t = 1 : T | \text{SK}(n).\text{EQ.0}) : \& \\ & \sum(i = 1 : M) a(n, i) * P(i, t, n) < W(n, t) \end{aligned} \quad (31)$$

モデル記述言語 mp-model の記述は、通常解くべき問題の記述が完了したことを示すとともに、実際に問題を解くソルバーである最適化モジュール mp-opt (mp-model と同様、これ以降、mp-opt は最適化モジュールを意味しているものとする) への入力データであるマトリックスファイルの生成を宣言する GENERATE セクションで終わる。

GENERATE (32)

上記に示したように、mp-model 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、これによって最適化モジュール mp-opt の入力データであるマトリックスファイルの生成を行うことが出来る。

計算手続きを記述する

次に、モデル記述言語を持つ数値計画ソフトウェアにおいて 3.2 項で述べた近似手続きを実現する

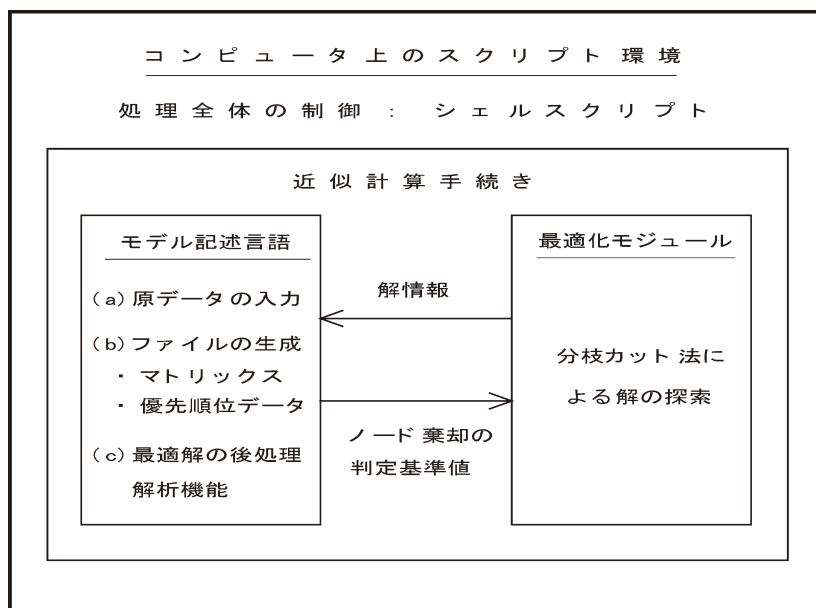


図6 モデル記述言語における解法システムの枠組み

計算手続きの記述について述べる。モデル記述言語を用いて近似手続きを実現するための解法システムの枠組みは図6に示す通りである⁽¹⁵⁾。

なお、図6の中で示されているファイルの生成にある優先順位データとは、3.2項で述べた分枝変数の選択における優先順位を指定するファイルのことであり、mp-modelでは以下のように記述する。

DIRECTIVES

$X(i=1:M, t=1:T, n=1:N | SK(n).EQ.1).PR.100$ (33)

$V0(i=1:M, n=1:N).PR.200$

$U0(i=1:M, n=1:N).PR.200$

なお、優先順位のデフォルト値は500であり、この値が小さい程分枝の優先度は高い。従って、(1) 段取り替えの回数を表す変数の優先順位：100、(2) 生産および引き取りの初期指示量を表す変数の優先順位：200、(3) 実際の生産量および引き取り量を表す変数の優先順位：500（デフォルト値）とした。

図6で示されている近似手続きはモデル記述言語 mp-model が持っている最適解の後処理解析機能⁽¹⁶⁾とオペレーティングシステム（OS）上のシェルスクリプト⁽¹⁷⁾による計算手順の制御によって実現した計算手続きである。以下では、近似手続きにおける mp-model の最適解の後処理解析機能の役割とシェルスクリプトで制御する計算手順について述べる。なおこれ以降、線形計画法の手続きを LP 手続き、線形計画問題を単に LP と表現する。また同様に、分枝限定法の手続きを IP 手続き、整数計画問題を単に IP と表現する。

(1) mp-model の最適解の後処理解析機能の役割

(a) LP 手続きと IP 手続きの区分：フラグを用いて分岐する

(b) 手続きの終了判定

: IP 手続きで、(1) 近似最適解が得られた場合、(2) 整数解が存在しない場合、または (3) LP 解が同時に整数解となっていた場合、手続き終了フラグを設定する。

(c) ノード棄却の判定基準値の設定

: IP 手続きで整数解が改善され手続きの継続が可能な場合、ノード棄却の判定基準値 CUTOFF を更新・設定し、ファイルに出力する (3.2 項で述べた式 (25) 参照)。

以上の役割をモデル記述言語 `mp-model` を用いて、`GENERATE` セクションの後に記述する。

(2) 近似手続きの計算手順

(a) 手順 1: `mp-model` で `mp-opt` の入力データとなるマトリックスファイルおよび分枝変数の選択のために優先順位データファイルを生成し、手順 2 へ。

(b) 手順 2: `mp-opt` で LP 解を求め基底情報を保存しておく。

ここで、LP 解があれば手順 3 へ。

LP 解がなければ手続きを終了する。

(c) 手順 3: 近似手続きの計算戦略に従い、`mp-opt` で IP の初期解を探索し解情報を保存した後、手順 4 へ。

(d) 手順 4: `mp-model` が設定した手続き終了フラグにより手続きの終了判定を行う。

ここで、終了判定の場合は手続きを終了する。

手続き継続の場合は CUTOFF を更新し、手順 5 へ。

なお、手続き終了のケースは次の通り。

ケース 1: 近似最適解が得られた。つまり最新の CUTOFF で枝の探索を完了し近似最適判定ができた場合。

ケース 2: 整数解は存在しない場合。

ケース 3: LP 解が同時に整数解となっていた場合。

(e) 手順 5: 前回の `mp-opt` の解情報を回復し、`mp-model` が更新・設定している CUTOFF を用いて、`mp-opt` で整数解を探索し解情報を保存した後、手順 4 へ。

モデル記述言語による解法システムは、モデル記述言語 `mp-model` と最適化モジュール `mp-opt` 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 6 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る。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) と計算手続きの記述 (Procedure Describing) を一つの環境で実現しているということができる。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。

4.3 モデリングシステムにおける記述

Kallrath [78] 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

リッ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。

- (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
- (4) 分枝限定 Tree およびマトリックスファイルの表示

以下、本項では2節で提示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 FICO Xpress の Xpress-Mosel [68] (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。なお Mosel によるモデルの記述方法の詳細は以下に示すように、モデル記述言語 mp-model における記述方法と異なるが、モデル記述機能そのものは mp-model の機能を継承している。このため、以下ではモデル記述の後、どのようにノード棄却の判定基準値を設定し近似手続きを実現しているかについて述べる。

なお、前項において mp-model による記述例で示した生産能力による生産量制約を表す式 (12) をモデリングシステム Mosel で記述すると以下ようになる。

! Constraints (12)

$$\begin{aligned} & \text{forall } (n \text{ in } 1..N, t \text{ in } 1..T \mid SK(n)=0) \\ & \quad C12(n, t) := \text{sum}(i \text{ in } 1..M) a(n, i) * P(i, t, n) <= W(n, t) \end{aligned} \quad (34)$$

Mosel では、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。

```

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

```

(35)

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 (例えば 0.01 に設定)。

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 (get-

param)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。

これらの記述の後、Mosel は以下のようなコマンドを発行しソルバーである Optimizer に求解の指示を与える。式 (36) は callback 機能といわれるもので、整数解が得られた時点で求解を一時停止し、式 (35) 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 (37) の 3 行目は、計算時間が 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 (38) 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') (36)
```

```
setparam ('XPRS_loadnames', true)
setparam ('XPRS_verbose', true) (37)
setparam ('XPRS_maxtime', -600)
```

```
loadprob (OBJ1)
readdirs ('exdircut.dir') (38)
minimize (OBJ1)
```

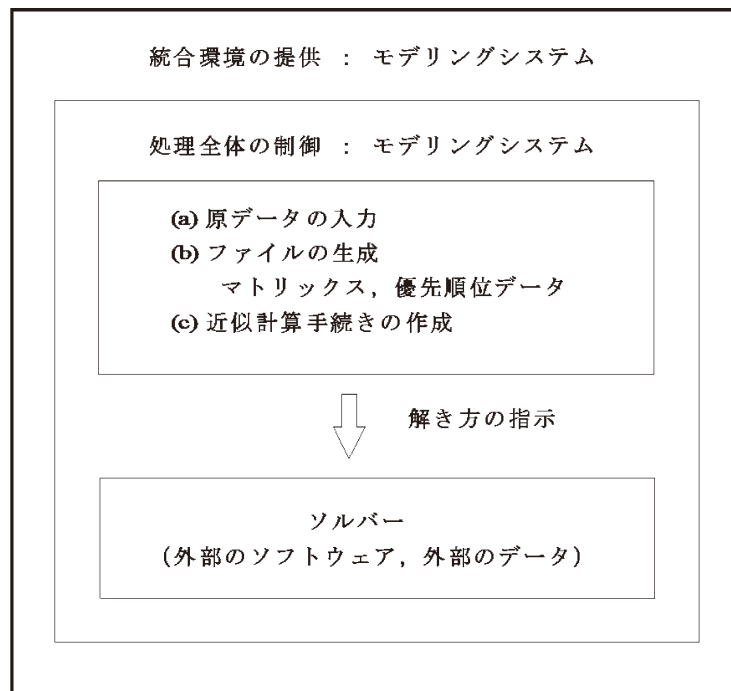


図7 モデリングシステムにおける解法システムの枠組み

図7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ならずソルバーに対してモデルの解き方を指示する形式となっている。具体的にはソルバー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ソルバーを制御している。上述の `setcallback`, `readdirs` 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) と計算手続きの記述 (Procedure Describing) 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。

5. 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、数理計画ソフトウェアにおけるモデルと計算手続きの記述に関する変遷を概観するとともに、それらの課題について述べた。まず汎用数理計画ソフトウェア MPS を用いて、短い計算時間で近似最適解を求めるための計算手続きの記述について述べた。次にユーザインターフェースの改善の観点から1970年代末以降開発が進められ、1990年代以降日本国内においても入手が容易になってきたモデル記述言語を用いたモデルの記述と近似手続きの実現方法を示した。汎用数理計画ソフトウェアは MPS 手続き制御言語といわれる制御言語を備えており、本研究で用いたモデル記述言語を持った数理計画ソフトウェアと比較した場合、ソルバー単体の機能に限れば、汎用数理計画ソフトウェアが提供する機能の方が充実したものといえる。図5で示したように近似計算手続きが非常に簡明な記述で実現されている。ジョブ制御言語および手続き制御言語に関わる十分な知識が前提となるが、汎用数理計画ソフトウェアを用いて反復的目標計画法の計算手続きを実現した研究事例もある⁽¹⁸⁾。一方、モデル記述言語を持った数理計画ソフトウェアにおいては、いわばソルバーの簡素な機能をモデル記述言語の豊富な記述力で補っているといった感がある。4.2項で示した計算手続きはモデル記述言語が持っている最適解の後処理解析機能およびその記述力によって実現できたものである。本稿で示したものは一つの事例であるが、数理計画ソフトウェアとしてモデル記述言語が持っている記述力およびその柔軟性には様々な利用可能性があるものと考えられ、数理計画法によるモデリングにおいてスプレッドシートと同様に、モデル記述言語の今後の動向は注目すべきものである。このモデル記述言語が2000年代以降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モデリングシステムという言葉が用いられている。4.3項においてこのモデリングシステムを用いた計算手続きの記述について述べた。その特徴を簡潔に表現するとすれば、モデルの記述 (Model Describing) と計算手続きの記述 (Procedure Describing) を一つの環境で実現していることができる。モデリングシステムを含め近年の数理計画ソフトウェアは、「モデルを記述しながら整数計画問題を早く解く」環境を提供しつつある。

数理計画法によるモデリングにおいて、チューニング機能も含めソルバー単体における求解性能の向上はもちろん重要であるが、本稿で示したモデリングシステムの活用も含め、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、解結果のフィードバックとその後の手順の指示および計算結果レポートの出力などを含めた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。従って、本研究で主に使用してきた FICO Xpress と同様、今後 Gurobi Optimizer [73] など他の数値計画ソフトウェアを用いる場合においても、モデル記述言語、モデリングシステムおよび各種ツールなどとの連携についての検証も、引き続き実施していきたいと考えている。

謝 辞

本稿は平成 27 年度専修大学研究助成・個別研究・研究課題「生産計画モデルを対象とした数値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、「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 [2], 平木 [11], ジャストインタイム生産システム研究会 [15], 小谷 [23], 黒田他 [25], 門田 [30], 村松 [31], 日本生産管理学会編 [33], 大野 [34], 大野 [35] および大野監修-門田編著 [36]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, 藤江 [7], 茨木 [12], [13], 茨木-福島 [14], 今野 [20], 今野-鈴木編 [22], 久保 [24], Beasley (ed.) [60], Carter-Price [64], Martin [86], Nemhauser-Wolsey [88] および Wolsey [97]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今野 [21], 宮代 [27], 宮代-松井 [28], 柳浦-野々部 [54], Achterberg [56], Bixby [62], Chen-Batson-Dang [65], Johnson-Nemhauser-Savelsbergh [77], Linderth-Savelsbergh [80] および Lodi [84]などを参照するとよい。
- (3) モデル記述言語を含め数値計画ソフトウェアの近年の進展については, Berthold et al. [3], 藤江 [8], Atamtürk-Savelsbergh [57], Kallrath (ed.) [78], [79], Linderth-Ralphs [81], Linderth-Lodi [82] および OR/MS Today 誌に掲載される Software Surveys [89] (例えば, Fourer [69]-[71])などを参照するとよい。なおソルバーの並列処理機能の動向については, 品野-藤江 [38], 品野-Achterberg-藤江 [39], Barney [58], Talbi (ed.) [92] および Xu et al. [98]などを参照するとよい。またソルバーにおけるチューニング機能については, Baz-Hunsaker-Prokopyev [59], Fourer [69], Hutter-Hoos-Leyton-Brown [74] および Lodi [84]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 [43], 渡辺-安-平木 [50]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 [43], Watanabe-Hiraki [95]を参照。
- (6) Optimizer の過去のバージョンにおいては, ノードの選択は以下のような下界値優先則と奥行き優先則との折衷則が初期設定として採用されていた。
 - ・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。
 - ・なお良いノードとは, 選択の対象となっている子問題の中で最良の下限値 (最小化問題の場合) を持つノードとする。
- (7) 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 [12], 今野-鈴木編 [22]などを参照するとよい。
- (8) ノード棄却の判定基準値の初期設定は式 (24) で示される通りであるが, 本稿が対象としている数値計画モデルの場合, その評価関数値は整数となるため, 実際に実行される計算においては, Optimizer は自動的に $CUTOFF = IPOBJ - 1$ と設定する。
- (9) 段取り替えの詳細については門田 [30]を参照するとよい。
- (10) 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 [43] - [48], Watanabe-Hiraki [94], [95]において詳細な数値検証を加えている。

- (11) 汎用数値計画ソフトウェア MPS については、反町編 [40]、反町他 [41]などを参照するとよい。また著者らが数値計算で実際に使用した MPS については、日本電気 [32]を参照。
- (12) MPS フォーマットについては、前田 [26]、反町編 [40]、渡辺 [43] および Williams [96]などを参照するとよい。また Linear Programming FAQ のホームページ [83] 上にも MPS フォーマットについての解説がある。
- (13) オペレーティングシステム (OS)、ジョブ制御言語 (JCL) については、阿江 [1]、浦・市川編 [42]などを参照するとよい。また実際に使用した MPS での JCL および手続き制御言語の詳細については日本電気 [32]を参照。
- (14) モデル記述言語の詳細については前田 [26]、渡辺 [43]、Kallrath (ed.) [78]、[79]、Sharda-Rampal [91] および Williams [96]などを参照するとよい。また Linear Programming FAQ のホームページ [83] 上にもモデル記述言語についての解説がある。
- (15) モデル記述言語における解くべき整数計画問題に対する計算手続きの記述の詳細については、渡辺 [43]、[44]、渡辺-錦織-平木 [51]、渡辺-宇佐美 [52]、Watanabe [93]を参照。
- (16) 使用したモデル記述言語 mp-model が持っている最適解の後処理解析機能の手続き上の概要は以下の通り。なお詳細については Dash Associates [67]、渡辺 [43]を参照。
・モデル記述の終了を示す GENERATE セクション以降にさらに記述があると、mp-model は自動的にモデルの状態を、問題名に拡張子 .svm を付けたファイルに保存する。最適化モジュール mp-opt で最適化した後、mp-model を再び実行し、ここで restore コマンドを発行すると mp-opt で得た最適解の情報を読み込むことができる状態となる。ここでさらに input コマンドを発行すると GENERATE セクション以降に記述されている文が実行される。
- (17) シェルスクリプトを含めた UNIX 全般については、例えば林 [10]、鬼頭 [19]などを参照。
- (18) 汎用数値計画ソフトウェア上の手続き制御言語を用いて反復的目標計画法の計算手続きを記述した例としては、例えば Crowder-Sposito [66]、Ignizio-Perlis [75]、Markland-Vickery [85]などを参照するとよい。

参考文献

- [1] 阿江忠：「計算機システム」、オーム社、東京 (1987)。
- [2] 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)。
- [3] Berthold, T., Gleixner, A.M., Heinz, S., Koch, T. and Shinano, Y., “SCIP Optimization Suite を利用した混合整数 (線形 / 非線形) 計画問題の解法”, 第 24 回 RAMP シンポジウム論文集, pp. 165-192 (2012)。
- [4] 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)。
- [5] Bixby, R.E., “Progress in Optimization & Gurobi Optimizer”, Gurobi Optimizer 販売記念特別講演会資料 (2010)。
- [6] Bixby, R.E., Gu, Z. and Rothberg, E., “Presolve for Linear and Mixed-Integer Programming”, 第 24 回 RAMP シンポジウム論文集, pp. 193-200 (2012)。
- [7] 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)。
- [8] 藤江哲也, “最近の混合整数計画ソルバーの進展について”, オペレーションズ・リサーチ, Vol. 56, No. 5, pp. 263-268 (2011)。
- [9] 藤江哲也, “整数計画法による定式化入門”, オペレーションズ・リサーチ, Vol. 57, No. 4, pp. 190-197 (2012)。
- [10] 林晴比古, 「新 Linux/UNIX 入門」, ソフトバンクパブリッシング, 東京 (2000)。
- [11] 平木秀作, 「自動車の現地生産と部品調達」, 溪水社, 広島 (1996)。
- [12] 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)。
- [13] 茨木俊秀, 「最適化の数学」, 共立出版, 東京 (2011)。
- [14] 茨木俊秀, 福島雅夫, 「最適化の手法」, 共立出版, 東京 (1993)。
- [15] ジャストインタイム生産システム研究会編, 「ジャストインタイム生産システム」, 日本工業新聞社, 東京 (2004)。
- [16] 株式会社オクトーバー・スカイ, Gurobi Optimizer ソリユーションセミナー 2012 資料 (2012)。
- [17] 株式会社オクトーバー・スカイ, Gurobi Optimizer ソリユーションセミナー 2015 資料 (2015)。
- [18] 株式会社オクトーバー・スカイ, Gurobi Optimizer ソリユーションセミナー 2016 資料 (2016)。
- [19] 鬼頭繁治, 「ユーザのための UNIX」, サイエンス社, 東京 (1993)。

- [20] 今野浩, 「整数計画法」, 産業図書, 東京 (1981).
- [21] 今野浩, 「役にたつ一次式—整数計画法「気まぐれな王女」の50年—」, 日本評論社, 東京 (2005).
- [22] 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).
- [23] 小谷重徳, 「理論から手法まできちんとわかるトヨタ生産方式」, 日本工業新聞社, 東京 (2008).
- [24] 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).
- [25] 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).
- [26] 前田英次郎, “数値計画支援システム”, 第5回 RAMP シンポジウム論文集, pp. 57-72 (1993).
- [27] 宮代隆平, “ここまで解ける整数計画—近年の発展—”, 第20回 RAMP シンポジウム論文集, pp. 1-21 (2008).
- [28] 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).
- [29] 宮崎知明, “数値計画法システム進化の歴史と今後の方向—All-in-One ソルバに近づく LocalSolver—”, オペレーションズ・リサーチ, Vol. 61, No. 1, pp. 35-42 (2016).
- [30] 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).
- [31] 村松林太郎, 「新版生産管理の基礎」, 国元書房, 東京 (1979).
- [32] 日本電気, 「数値計画システム MPS/EXA 利用の手引<機能操作編>」, 東京 (1991).
- [33] 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).
- [34] 大野勝久, 「Excel による生産管理—需要予測, 在庫管理から JIT まで—」, 朝倉書店, 東京 (2011).
- [35] 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).
- [36] 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).
- [37] 品野勇治, “最適化研究における数値実験を中心としたアプリケーション駆動研究サイクル”, 日本オペレーションズ・リサーチ学会関西支部シンポジウム「異分野コミュニケーションによる最適化の広がり」講演会資料 (2013).
- [38] 品野勇治, 藤江哲也, “混合整数計画ソルバーの並列化”, オペレーションズ・リサーチ, Vol. 52, No. 10, pp. 633-638 (2007).
- [39] 品野勇治, Achterberg, T., 藤江哲也, “混合整数計画ソルバの並列化について”, 第20回 RAMP シンポジウム論文集, pp. 23-43 (2008).
- [40] 反町洋一編, 「線形計画法の実際」, 産業図書, 東京 (1992).
- [41] 反町洋一, 小田稔周, 熊野長次郎, “汎用数値計画ソフトウェア”, オペレーションズ・リサーチ, Vol. 38, No. 3, pp. 113-123 (1993).
- [42] 浦昭二, 市川照久共編, 「情報処理システム入門 第2版」, サイエンス社, 東京 (1998).
- [43] 渡辺展男, 「多段階生産・在庫・運搬システム—数値計画法によるモデリング—」, 溪水社, 広島 (1999).
- [44] 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).
- [45] 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巻, 第1・2号, pp. 13-29 (2001).
- [46] 渡辺展男, “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—”, 専修経営研究年報, No. 29, pp. 27-55 (2005).
- [47] 渡辺展男, “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—”, 専修経営研究年報, No. 35, pp. 1-28 (2011).
- [48] 渡辺展男, “数値計画ソフトウェアの進展について”, 専修大学情報科学研究所 情報科学研究, No. 33, pp. 21-42 (2013).
- [49] 渡辺展男, “チューニング機能を活用した整数計画問題の解法(2)—Gurobi Optimizerを用いた数値実験—”, 専修大学情報科学研究所 情報科学研究, No. 34, pp. 39-54 (2014).
- [50] 渡辺展男, 安范俊, 平木秀作, “引っ張り型生産指示方式の数値計画的アプローチ”, 日本経営工学会誌, Vol. 44, No. 6, pp. 478-486 (1994).
- [51] 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, 数値計画モデルの応用—構築と解法と分析—, pp. 14-28, 日本 OR 学会 (1995).
- [52] 渡辺展男, 宇佐美嘉弘, “数値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値計画モデル—”, 専修大学情報科学研究所 所報, No. 68, pp. 1-25 (2008).
- [53] 渡辺展男, 宇佐美嘉弘, “数値計画ソフトウェアを用いた整数計画問題の解法 (2) —並列処理機能とチュー

- ニシグ機能の効果—”, 専修大学情報科学研究所 情報科学研究, No. 32, pp. 17-39 (2012).
- [54] 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).
- [55] 山下浩, 蒲地政文, 畔上秀幸, 斉藤努, 枇々木規雄, 滝根哲哉, 金森敬文, 「モデリングの諸相—OR と数理科学の交叉点—」, 近代科学社, 東京 (2016).
- [56] Achterberg, T., Constraint Integer Programming, Ph. D. Thesis, Technische Universität Berlin (2007).
- [57] Atamtürk, A. and Savelsbergh, M.W.P., “Integer-Programming Software Systems”, *Annals of Operations Research*, Vol. 140, pp. 67-124 (2005).
- [58] Barney, B., Introduction to Parallel Computing, http://computing.llnl.gov/tutorials/parallel_comp/
- [59] Baz, M., Hunsaker, B. and Prokopyev, O., “How Much Do We “Pay” for Using Default Parameters?”, *Computational Optimization and Applications*, Vol. 48, No. 1, pp. 91-108 (2011).
- [60] Beasley, J.E. (ed.), *Advances in Linear and Integer Programming*, Oxford University Press, Oxford (1996).
- [61] Bixby, R.E., “Solving Real-World Linear Programs : A Decade and More of Progress”, *Operations Research*, Vol. 50, No. 1, pp. 3-15 (2002).
- [62] Bixby, R.E., “A Brief History of Linear and Mixed-Integer Programming Computation”, in : Grötschel, M. (ed.) *Optimization Stories*, pp. 107-121 (2012).
- [63] Bixby, R.E., Fenelon, M., Gu, Z., Rothberg, E. and Wunderling, R., “Mixed-Integer Programming : A Progress Report”, in *The Sharpest Cut* (Grötschel, M. ed.), SIAM, Philadelphia, pp. 309-327 (2004).
- [64] Carter, M.W. and Price, C.C., *Operations Research : A Practical Introduction*, CRC Press, Boca Raton (2000).
- [65] Chen, D.S., Batson, R.G. and Dang, Y., *Applied Integer Programming : Modeling and Solution*, John Wiley & Sons, New Jersey (2010).
- [66] Crowder, L.J. and Sposito, V.A. , “Sequential Linear Goal Programming : Implementation via MPSX/370E”, *Computers and Operations Research*, Vol. 18, No. 3, pp. 291-296 (1991).
- [67] Dash Associate, XPRESS-MP Reference Manual Release 10 (1997).
- [68] FICO : Getting Started with Xpress Release 8.2 (2016) ;
Xpress-Mosel Reference Manual Release 4.4 (2017) ;
Xpress-Optimizer Reference Manual Release 31.01 (2017) ;
Xpress-Tuner User Guide Release 1.1 (2009).
- [69] Fourer, R., “Linear Programming Software Survey”, *OR/MS Today*, Vol. 40, No. 3, pp. 40-53 (2013).
- [70] Fourer, R., “Linear Programming Software Survey”, *OR/MS Today*, Vol. 42, No. 3, pp. 52-63 (2015).
- [71] Fourer, R., “Linear Programming Software Survey”, *OR/MS Today*, Vol. 44, No. 3, pp. 48-59 (2017).
- [72] Gurobi Optimization :
Gurobi Optimizer Reference Manual Version 5.5 (2013) ;
Webinar : “Recent Improvements in the Gurobi Optimizer” (April, 2013).
- [73] Gurobi Optimization :
Gurobi Optimizer Reference Manual Version 7.5 (2017) ;
Workshop : “Recent Developments in the Gurobi Optimizer” (October, 2017).
- [74] Hutter, F., Hoos, H.H., Leyton-Brown, K., “Automated Configuration of Mixed Integer Programming Solvers” in *Lecture Notes in Computer Science*, Vol. 6140 (Lodi, A., Milano, M. and Toth, P. eds.), pp. 186-202, Springer, Heidelberg (2010).
- [75] Ignizio, J.P. and Perlis, J.H., “Sequential Linear Goal Programming : Implementation via MPSX”, *Computers and Operations Research*, Vol. 6, pp. 141-145 (1979).
- [76] ILOG : ILOG CPLEX, 現在は IBM ILOG CPLEX Optimizer.
- [77] 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).
- [78] Kallrath, J. (ed.), *Modeling Languages in Mathematical Optimization*, Kluwer Academic Publishers, Massachusetts (2004).
- [79] Kallrath, J. (ed.), *Algebraic Modeling Systems : Modeling and Solving Real World Optimization Problems*, Springer, Heidelberg (2012).
- [80] Linderoth, J.T. and Savelsbergh, M.W.P., “A Computational Study of Search Strategies for Mixed Integer Program-

- ming”, *INFORMS Journal on Computing*, Vol. 11, No. 2, pp. 173-187 (1999).
- [81] Linderorth, J.T. and Ralphs, T.K., “Noncommercial Software for Mixed Integer Linear Programming”, in *Integer Programming : Theory and Practice* (Karlof, J.K. ed.), CRC Press, Boca Raton, pp. 253-303 (2005).
- [82] Linderorth, J.T. and Lodi, A., “MILP Software”, in *Wiley Encyclopedia of Operations Research and Management Science* (Cochran, J. ed.), Wiley, New York, Vol. 5, pp. 3239-3248 (2011).
- [83] Linear Programming FAQ, <http://neos-guide.org/content/lp-faq>
- [84] Lodi, A., “Mixed Integer Programming Computation”, in *50 Years of Integer Programming 1958-2008* (Jünger, M. et al. eds.), Springer, Heidelberg, pp. 619-645 (2010).
- [85] Markland, R.E. and Vickery, S.K., “The Efficient Computer Implementation of a Large-scale Integer Goal Programming Model”, *European Journal of Operational Research*, Vol. 26, pp. 341-354 (1986).
- [86] Martin, R.K., *Large Scale Linear and Integer Optimization : A Unified Approach*, Kluwer Academic Publishers, Massachusetts (1999).
- [87] 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).
- [88] Nemhauser, G.L. and Wolsey, L.A., *Integer and Combinatorial Optimization*, John Wiley & Sons, New York (1988).
- [89] OR/MS Today Software Surveys, <http://www.orms-today.org/ormsurveys.html>
- [90] Sarker, R.A. and Newton, C.S., *Optimization Modeling : A Practical Approach*, CRC Press, Boca Raton (2008).
- [91] Sharda, R. and Rampal, G., “Software Survey : Algebraic Modeling Languages on PCs”, *OR/MS Today*, Vol. 22, No. 3, pp. 58-63 (1995).
- [92] Talbi, E. (ed.), *Parallel Combinatorial Optimization*, John Wiley & Sons, Hoboken, NJ (2006).
- [93] 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).
- [94] 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).
- [95] Watanabe, N. and Hiraki, S., “A Modeling Approach to a JIT-based Ordering System”, *Annals of Operations Research*, Vol. 69, pp. 379-403 (1997).
- [96] Williams, H.P., *Model Building in Mathematical Programming* 5th ed., John Wiley & Sons, Chichester, England (2013).
- [97] Wolsey, L.A., *Integer Programming*, John Wiley & Sons, New York (1998).
- [98] Xu, Y., Ralphs, T.K., Ladányi, L. and Saltzman, M.J., “Computational Experience with a Software Framework for Parallel Integer Programming”, *INFORMS Journal on Computing*, Vol. 21, No. 3, pp. 383-397 (2009).