

数理計画ソフトウェアの進展について

渡 辺 展 男 (専修大学経営学部)

On the Progress in Mathematical Programming Software — Computational Experience with FICO Xpress —

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 problems to be solved are (1) to reduce the computational time because the model includes many integral variables, and (2) to improve an interface for the model solution because mathematical modeling has required transformations of the model form. In order to tackle the problems, this paper presents a solution procedure using a mathematical programming software suite (FICO Xpress) with a modeling system, a parallel MIP (Mixed Integer Programming) solver and a tuning tool for the parameters. The aim of this paper is two-fold. One is to illustrate the computational progress of the mathematical programming software and the other is to verify the effectiveness of the solution procedure, through numerical examples of the model applied to an automobile parts manufacturer.

キーワード： 整数計画法, 数理計画ソフトウェア, 並列処理, チューニング, かんばん方式

Key words: Integer Programming, Mathematical Programming Software, Parallel Computing, Tuning, 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数値実験の対象として、数理計画ソフトウェアの進展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。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 [69]。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・構築を行い、数値計算を通してその有効性を検証した [29]-[31], [35], [66], [68]。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 [32]。

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⁾。1990 年代以降、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、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], [3], [5], [40], [44]。例えば、代表的な数理計画ソフトウェアである ILOG CPLEX [51] 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 3,300 倍といわれている。これに同期間のマシンの計算速度の改善率として見込まれる数値 1,600 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], [43], [44]。この 530 万倍という数字は、過去において 2 ヶ月を要した計算が、1 秒で終了することを意味している。また混合整数計画法 (mixed-integer programming, MIP) における計算上の進展については、ILOG CPLEX から Gurobi Optimizer [49] の最新バージョン 5.0 にいたる 1991 年から現在までの改善率は、アルゴリズム部分だけで約 37 万 5 千倍といわれている。これに同期間のマシンの計算速度の改善率として見込まれる数値 2,000 倍を合わせ考えると、この期間での混合整数計画法の平均的な改善率は約 7 億 5 千万倍であるとの報告がある [4], [12]。この 7 億 5 千万倍という数字は、過去において 23 年以上を要した計算が、1 秒で終了することを意味する驚異的な数字である。このように数理計画法、特に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⁾。

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、ソルバーへ解法を指示 (解法の記述) そして数理計画モデルを用いた解決システムの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。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、著者らが従来から使用してきている FICO Xpress [47] による数値計算を通して、(1) 数理計画ソフトウェアの求解性能の向上について述べること、および (2) 本研究で提案している計算手続きの有効性、近年実装されるようになった並列処理機能ならびにチューニング機能の効果について得られた知見を整理し、数理計画法によるモデリングに関する今後の展望について述べることにあ

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 節において本研究で使

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 [29], [68]。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。

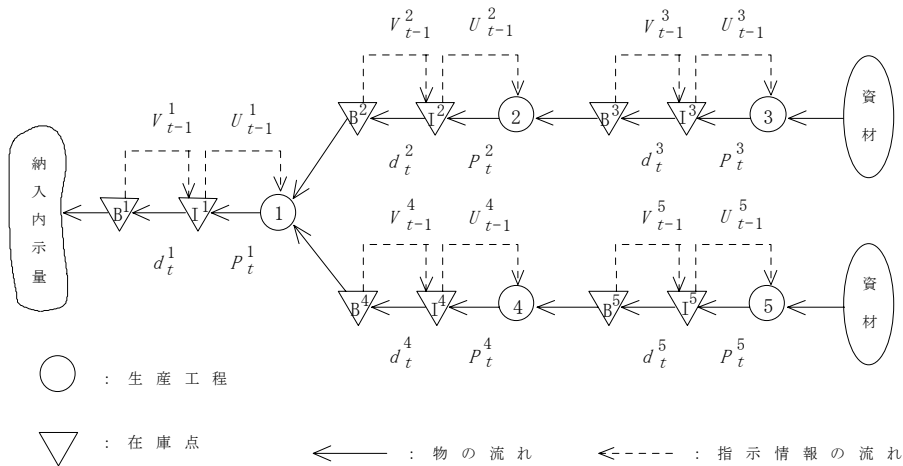
- (1) 一つの組立工程に収束していく多段工程で N 工程から構成されており、 $n \in \{1, 2, \dots, N\}$ で工程を表す。なお最終工程は $n=1$ とする。
- (2) 各工程は生産工程、加工済み在庫点および後続工程加工待ち在庫点 ($n=1$ では納入待ち製品在庫点) から成る。
- (3) 期間を $t \in \{0, 1, \dots, T\}$ で表す。計画期間は 1 期より始まり T 期で終了する。
またこのモデルは以下の条件で示される生産状況を対象としている。
- (1) 受注先から最終製品の各期の納入量についての内示があり、受注残は認められない。
- (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

図1 モデルの概念図の例 ($N=5$)

- 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) 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

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

subject to

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

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

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

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

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

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

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

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

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

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

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

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

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

ここで,

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

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

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

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

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

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

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

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

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

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 [61] 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 [47] (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 $X_i^{n(i)}$ 、(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⁽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⁽¹⁰⁾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）。

- (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. 数値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。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えることができる。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。

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 [47]（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⁽¹¹⁾。

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

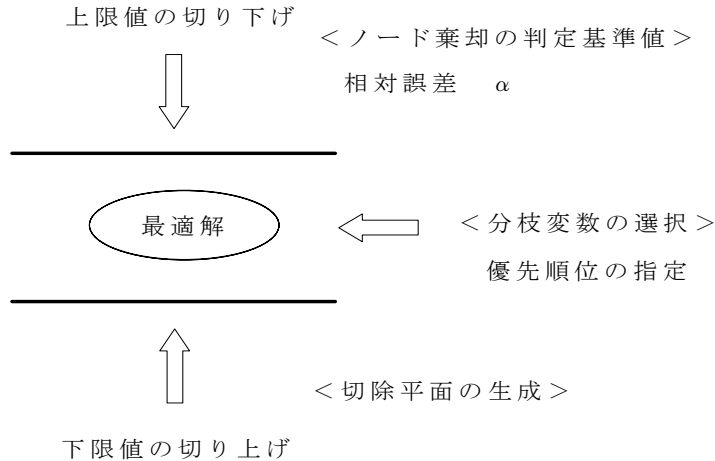


図2 近似手続きの枠組み

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

(26)

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(27)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(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)
readdirs ('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

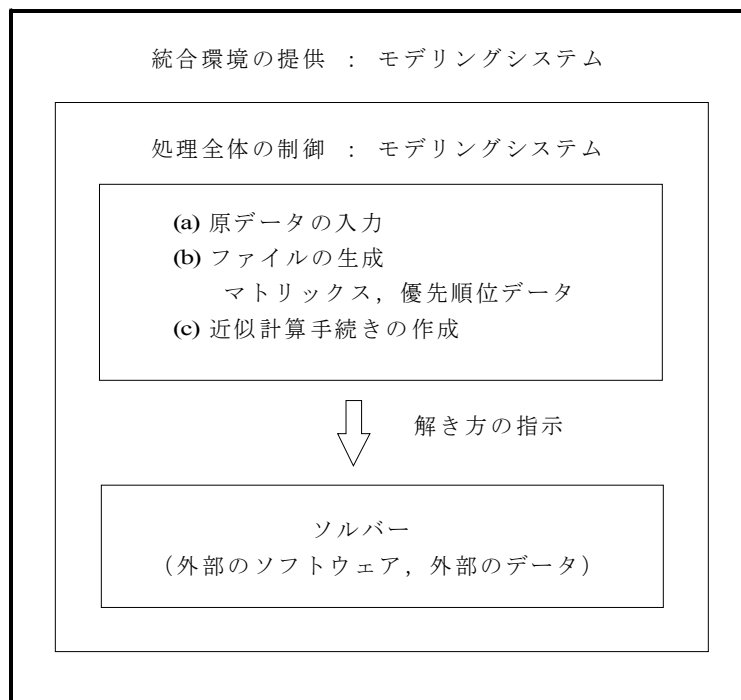


図 3 モデリングシステムを用いた解法の枠組み

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 並列処理機能およびチューニング機能

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。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いる。特に、本研究で使用している Xpress-Optimizer および Gurobi Optimizer など整数計画法における最新の解法を実装したソルバーにおける並列処理機能の導入は、求解性能の向上において有意義であると考えられる⁽¹²⁾。

今回実施した数値計算で使用した CPU は、2 コア、4 スレッドの Intel Core i7-620M であるが、Xpress-Optimizer および Gurobi Optimizer は、マルチスレッドであることを検知した後、それぞれが実装しているスレッドレベルの並列処理によって、分枝限定法および分枝カット法における分枝木の探索 (parallel tree search) を実行する。

またソルバーにおいては、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。多くの制御パラメータは問題の求解に対して柔軟性を与えてくれる一方で、それら制御パラメータの設定の組合せは複雑となり、適切な設定を手作業で探ることは困難となる。従って、そのチューニング作業を支援するツールの開発は数理計画法によるモデリングにおける一つの課題となっている [41], [48], [50], [59]。本研究で使用している Xpress-Optimizer には、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 170 件の制御パラメータがあるが、FICO Xpress においては Xpress-Tuner [47] (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], [3], [5], [40], [44]。

- (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⁽¹³⁾。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、Phase1 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、Phase2 として Phase1 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、Phase3 としてユーザが Tuner の実行を停止するまでの間、Phase1 から Phase2 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。

今回使用した Tuner においては、ソルバーが並列処理機能を実装したことにもない、Tuner の実行において並列処理を行うかどうかを選択できるようになっている（並列処理を行う場合は、Tune Parallel Xpress を選択する）。数値計算の結果は次節で述べるが、今回は 1 回の実行時間を 30 秒と定め、約 200 回の繰り返し計算を実行し、Phase3 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、並列処理機能および本研究で使用し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 の旧バージョンと最新バージョンを用いた数値計算を行い、ソルバーの求解性能の向上を示すとともに、並列処理機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

本研究で提案する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 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（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

$$\begin{aligned} D_t^{(1)} &= 20, D_t^{(2)} = 15, D_t^{(3)} = 5 & (t = 1, 10), \\ D_t^{(1)} &= 30, D_t^{(2)} = 25, D_t^{(3)} = 5 & (t = 2, 3, \dots, 9). \end{aligned}$$

- (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

$$\begin{aligned} a^{n(i)} &= 6 \text{ 分} & (i = 1, 2, 3; n = 1, 2), \\ a^{n(i)} &= 3 \text{ 分} & (i = 1, 2, 3; n = 3, 4, 5). \end{aligned}$$

(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

$$\begin{aligned} B_0^{n(1)} &= 14, B_0^{n(2)} = 12, B_0^{n(3)} = 5 \\ I_0^{n(1)} &= 14, I_0^{n(2)} = 12, I_0^{n(3)} = 5 & (n = 1, 2, \dots, 5). \end{aligned}$$

(g) 期末目標在庫量

$$\begin{aligned} SB_t^{n(1)} &= 10, SB_t^{n(2)} = 8, SB_t^{n(3)} = 3 \\ SI_t^{n(1)} &= 10, SI_t^{n(2)} = 8, SI_t^{n(3)} = 3 & (n = 1, 2, \dots, 5; t = 1, 2, \dots, 10). \end{aligned}$$

(h) 生産仕掛量

$$\begin{aligned} P_0^{1(1)} &= 25, P_0^{1(2)} = 20, P_0^{1(3)} = 5 \\ P_0^{2(1)} &= 30, P_0^{2(2)} = 20, P_0^{2(3)} = 0. \end{aligned}$$

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	Intel Core i7-620M 2.66GHz
(2) RAM	4 GB
(3) オペレーティングシステム (OS)	Windows 7 Professional 32bit SP1
(4) 数理計画ソフトウェア	FICO Xpress Release 7.3
モデリングシステム	Xpress-Mosel Version 3.4.0
ソルバー	Xpress-Optimizer Version 23.01.05
統合環境	Xpress-IVE Version 1.23.00
チューニングツール	Xpress-Tuner Version 1.1.6

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。まずソルバーの求解性能の向上、計算手続きの有効性および並列処理機能の効果について考察を行うために行った FICO Xpress の旧バージョン Release2003G と最新バージョン Release7.3 を用いた数値計算の結果を表 1、表 2 および表 3 に示す。なお今回の数値計算で旧バージョンとして用いた Release2003G は、最終結果として計算打ち切りとなるものの 1994 年以来長年使用してきたこのソルバーにおいて、3,600 秒以内で最適解に到達した初めてのバージョンであり、その時点（2003 年）においてソルバーの求解性能の進展を実

表 1 計算結果 (1)

	Rel. 2003G 標準手続き	Xpress 7.3 標準手続き 並列処理なし	Xpress 7.3 標準手続き 並列処理あり
最初の整数解 Time 生成ノード 評価関数値	1 秒 138 634	3 秒 41 607	3 秒 104 598
最良の整数解 Time 生成ノード 評価関数値	13 回目 186 秒 68,714 561	18 回目 26 秒 5,399 561	10 回目 23 秒 7,697 561
最終結果 Time 生成ノード 評価関数値 判 定	3,600 秒 3,411,300 561 計算打ち切り	185 秒 50,991 561 最適解判定 探索完了	104 秒 55,707 561 最適解判定 探索完了

表 2 計算結果 (2)

	Rel. 2003G 標準手続き	Rel. 2003G 優先順位	Rel. 2003G 近似手続き
最初の整数解 Time 生成ノード 評価関数値	1 秒 138 634	0 秒 100 594	0 秒 100 594
最良の整数解 Time 生成ノード 評価関数値	13 回目 186 秒 68,714 561	13 回目 201 秒 124,606 561	5 回目 29 秒 16,252 563
最終結果 Time 生成ノード 評価関数値 判 定	3,600 秒 3,411,300 561 計算打ち切り	211 秒 146,109 561 最適解判定 探索完了	66 秒 40,916 563 相対誤差 0.01 探索完了

感したバージョンである。

表の見方は次の通り。

- (1) 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に, ヒューリスティクス, 切除平面の導入等で整数解が得られる場合もあり, その場合は, 最初の整数解の欄に (1 回目) と表記した。

表 1 はソルバーの初期設定に基づく標準手続きの計算結果である。最新バージョンでは, 並列処理を実施しない場合と実施した場合の二通りの計算を行っている。なお旧バージョンには並列処理機能はない。表 1 で示されているように, ソルバーの求解性能が著しく向上していること, および並列処理が効果的に機能していることは明らかである。

表 3 計算結果 (3)

	Xpress 7.3 標準手続き 並列処理あり	Xpress 7.3 優先順位 並列処理あり	Xpress 7.3 近似手続き 並列処理あり
最初の整数解 Time 生成ノード 評価関数値	3 秒 104 598	3 秒 174 606	3 秒 174 606
最良の整数解 Time 生成ノード 評価関数値	10 回目 23 秒 7,697 561	12 回目 20 秒 7,217 561	5 回目 17 秒 5,390 563
最終結果 Time 生成ノード 評価関数値 判 定	104 秒 55,707 561 最適解判定 探索完了	85 秒 45,955 561 最適解判定 探索完了	49 秒 24,441 563 相対誤差 0.01 探索完了

- (1) 並列処理を実施しない最新バージョン Release7.3 は旧バージョン Release2003G と比べ、 $1/7$ 以下の計算時間 (26 秒と 186 秒) と $1/12$ 以下の生成ノード数 (5,399 と 68,714) で最適解に到達している。また 3,600 秒時点で計算打ち切りとなっている旧バージョンに対して、185 秒の時点で最適判定を行い解の探索を完了している。
- (2) 並列処理を実施した場合の効果については、最適解への到達時点においては並列処理の有無で大きな違い (23 秒と 26 秒) は見られないものの、最適判定を行い解の探索を完了する時点においては 4 割以上の計算時間の削減 (104 秒と 185 秒) となっている。

表 2 および表 3 は本研究で提案している計算手続きの有効性について検討するために実施した旧バージョンと最新バージョンによる数値計算の結果である。なお最新バージョンにおいては、いずれも並列処理を実施している場合の計算結果である。表 2 および表 3 で示されているように、優先順位を用いた計算手続きおよび近似手続きは、解の探索を完了させることにおいて計算量削減の効果が見られる。あわせて表 2 と表 3 を比較することで明らかなことは、これら計算量削減の効果は最新バージョンにおいては、それほど顕著ではないということであろう。

- (1) 旧バージョン Release2003G の場合、優先順位を用いた計算手続きは標準手続きと比べ、最適解への到達時間はむしろ劣っている (201 秒と 186 秒) もの、3,600 秒時点で計算打ち切りとなっている標準手続きに対して、211 秒の時点で最適判定を行い解の探索を完了している。また最終結果における生成ノード数も大幅な削減 (146,109 と 3,411,300) となっている。
- (2) 旧バージョン Release2003G の場合、近似手続きは標準手続きと比べ、 $1/6$ 以下の計算時間 (29 秒と 186 秒) で最良解に到達し、3,600 秒時点で計算打ち切りとなっている標準手続きに対して、わずか 66 秒の時点で近似最適判定を行い解の探索を完了している。また最終結果における生成ノード数も大幅な削減 (40,916 と 3,411,300) となっている。
- (3) 旧バージョン Release2003G の場合、近似手続きは優先順位を用いた計算手続きと比べ、約 $1/7$ の計算時間 (29 秒と 201 秒) で最良解に到達し、 $1/3$ 以下の計算時間 (66 秒と 211 秒) で近似最適判定を行い解の探索を完了している。また最終結果における生成ノード数も $1/3$ 以下 (40,916 と 146,109) である。
- (4) 最新バージョン Release7.3 の場合、優先順位を用いた計算手続きは標準手続きと比べ、最適解

への到達時間はほぼ互角（20 秒と 23 秒）であるが、最適判定を行い解の探索を完了する時点においては約 2 割の計算時間の削減（85 秒と 104 秒）となっている。

- (5) 最新バージョン Release7.3 の場合、近似手続きは標準手続きと比べ、約 2 割の計算時間の削減（17 秒と 23 秒）で最良解に到達し、1/2 以下の計算時間（49 秒と 104 秒）で近似最適判定を行い解の探索を完了している。
- (6) 最新バージョン Release7.3 の場合、近似手続きは優先順位を用いた計算手続きと比べ、わずかに計算量を削減（17 秒と 20 秒）しており、約 4 割削減した計算時間（49 秒と 85 秒）で近似最適判定を行い解の探索を完了している。

表 1、表 2 および表 3 で示した数値実験を通して得られた知見は以下の通りである。

- (1) ソルバーの求解性能が著しく向上しており、あわせて並列処理が効果的に機能している。
- (2) 本研究でこれまで提案してきた計算手続きの有効性については、解の探索を完了させることにおいて計算量削減の効果が見られるものの、最新バージョン Release7.3 においては、それほど顕著ではない。

本稿で対象としている整数計画問題の規模程度であれば、近年の数値計画ソフトウェアにみられる求解性能の向上によって計算量の削減が十分可能な段階にまで至っていると判断されるため、計算手続きの有効性の検証については、対象とする問題の規模を大きくした場合の数値実験などを行うことが必要と考えられる。

次に、表 3 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および分枝変数の選択に関わる優先順位情報を用いて、Tuner の最新リリースを実行して得られた結果を述べる。標準手続きおよび優先順位の計算手続きにおいて、最適解が 30 秒以内で得られていることから、Tuner の 1 回の実行時間を 30 秒と定め、約 200 回の繰り返し実行を行った。なおソルバーが並列処理機能を実装したこととともない、Tuner の実行において並列処理を行うかどうかを選択できるようになっており、今回、Tune Parallel Xpress を選択し、並列処理をしながらチューニング作業を実施した。その結果チューニング機能の効果を検証するために行う数値計算においては、以下の制御パラメータの組み合わせを用いることとした。いずれも切除平面（cut）に関わる制御パラメータである。

$$\text{LNPITERLIMIT} = 50 \quad (30)$$

$$\text{TRECOVERCUTS} = 0 \quad (31)$$

$$\text{CUTFREQ} = 5 \quad (32)$$

Tuner を用いた 1 回 30 秒の計算において、最適値 561 に到達し計算を終了したのは、約 200 回の繰り返し実行の中で 37 件であった。その中で、多くの件数で現れた上位三つの制御パラメータを採用した。具体的には式 (30) の設定は 34 件で現れ、式 (31) の設定は 29 件そして式 (32) の設定は 23 件で現れた。式 (30) の設定は、Lift-and-Project Cut の改善に関わる回数は 50 回であることを意味している。また式 (31) の設定は、ルートノード以外の分枝木の探索（tree search）における Lifted Cover Inequality 生成時の round 数が 0、つまり生成しないことを意味している。なお 1 round とは、カットを生成した後に線形計画問題を解く 1 サイクルのことをいう。また式 (32) の設定は、分枝木の探索における切除平面の生成頻度が 5 であることを意味している。式 (30)－式 (32) の制御パラメータの設定で行った数値計算の結果を表 4 に示す。なお、表 4 で示した計算では分枝限定法の手続きに入る前にヒューリスティクスで整数解が得られているため、最初の整数解の欄に（1 回目）と表記している。

チューニングツールの活用が有効であることは、表 3 と表 4 の結果を比較することで明らかである。

表4 計算結果 (4)

Xpress 7.3 の 計算結果	Tuner 実行後 標準手続き 並列処理あり	Tuner 実行後 優先順位 並列処理あり	Tuner 実行後 近似手続き 並列処理あり
最初の整数解 Time 生成ノード 評価関数値	(1 回目) 1 秒 1,859	(1 回目) 1 秒 1,859	(1 回目) 1 秒 1,859
最良の整数解 Time 生成ノード 評価関数値	9 回目 16 秒 11,597 561	20 回目 5 秒 2,976 561	8 回目 13 秒 9,645 561
最終結果 Time 生成ノード 評価関数値 判 定	25 秒 21,029 561 最適解判定 探索完了	16 秒 13,677 561 最適解判定 探索完了	*最適解到達* 13 秒 11,397 561 相対誤差 0.01 探索完了

なお今回の Tuner の実行においては、MPS フォーマットの問題ファイルに加え分枝変数の選択に関わる優先順位情報も用いているため、優先順位の計算手続きにおいて制御パラメータ設定の効果が顕著に表れている。

- (1) 標準手続きにおいて、Tuner の実行で得られた制御パラメータを設定すると通常の標準手続きと比べ、約 3 割の計算時間の削減（16 秒と 23 秒）で最適解に到達し、1/4 以下の計算時間（25 秒と 104 秒）で最適判定を行い解の探索を完了している。
- (2) 優先順位の計算手続きでは、Tuner の実行で得られた制御パラメータを設定すると通常の優先順位の計算手続きと比べ、1/4 の計算時間（5 秒と 20 秒）で最適解に到達し、1/5 以下の計算時間（16 秒と 85 秒）で最適判定を行い解の探索を完了している。
- (3) 近似手続きにおいて、Tuner の実行で得られた制御パラメータを設定すると通常の近似手続きと比べ、約 2 割の計算時間の削減（13 秒と 17 秒）で近似最適解ではなく最適解に到達し、1/3 以下の計算時間（13 秒と 49 秒）で最適判定を行い解の探索を完了している。
- (4) 近似手続きと優先順位の計算手続きを比べると、近似手続きは優先順位の計算手続きよりも早く（13 秒と 16 秒）最適判定を行い解の探索を完了していることは事実であるが、最適解への到達そのものは優先順位の計算手続きよりも遅く（13 秒と 5 秒）なっており、近似手続きの有効性を判断することは難しい。

今回、前処理およびヒューリスティクスの効果を検証するために追加的な数値計算を実施した。具体的には、表 4 で示した計算結果のうち優先順位の計算手続きにおいて、ヒューリスティクスを適用しない場合と前処理を実施しない場合の数値計算を実施した。これらの計算結果を表 5 に示す。表 5 で示されている結果より、対象としている問題においては、ヒューリスティクスと比べ、前処理の実施が効果的であると判断される。

- (1) 最適解へ到達する時点および最適判定を行い探索完了する時点のいずれにおいても、ヒューリスティクスを適用する場合と適用しない場合に大きな差は見られない。
- (2) 前処理を実施しなかった場合、前処理を実施した場合と比べ、2 倍以上の計算時間（12 秒と 5 秒）で最適解への到達し、最適判定を行い解の探索を完了させるまでに、3 倍以上の計算時間（49 秒と 16 秒）を要している。

表 5 計算結果 (5)

Xpress 7.3 Tuner 実行後の 計算結果	優先順位 並列処理あり	優先順位 並列処理あり no heuristics	優先順位 並列処理あり no presolve
最初の整数解 Time 生成ノード 評価関数値	(1 回目) 1 秒 1,859	2 秒 162 617	2 秒 73 682
最良の整数解 Time 生成ノード 評価関数値	20 回目 5 秒 2,976 561	5 回目 5 秒 2,958 561	12 回目 12 秒 8,936 561
最終結果 Time 生成ノード 評価関数値 判 定	16 秒 13,677 561 最適解判定 探索完了	15 秒 13,717 561 最適解判定 探索完了	49 秒 46,341 561 最適解判定 探索完了

既に述べたように、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、切除平面 (cut)、前処理 (presolve)、分枝変数の選択 (variable selection)、ヒューリスティクス (heuristics)、およびノードの選択 (node selection) などにおける複数の工夫の積み重ねの成果といわれている [2], [3], [5], [40], [44]。このうち線形計画問題に対する解法を除いたこれら工夫の効果を比較した場合、効果があるのは順に、(1) 切除平面、(2) 前処理そして (3) 分枝変数の選択といわれている [4]。本稿で示した表 1 から表 5 までの計算結果は、ほぼそれと対応した結果を表していると考えられる。

なお本研究で長年提案してきた計算手続きは、分枝変数の選択に優先順位を導入するというものである。具体的には、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⁽¹⁰⁾。

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 4 の優先順位の計算手続きにおいて得られている値である。

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

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。

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、数理計画ソフトウェアの求解性能の向上、本研究で提案している計算手続きの有効性そして近年実装されるようになった並列処理機能ならびにチューニング機能の効果について、著者らが従来から使用してきている FICO Xpress による数値実験の結果から得られた知見を述べた。長年使用してきたソフトウェアの旧バージョンと最新バージョンを、同一の計算環境で実際に計算したことで、あらためて求解性能の進展を実感した。また追加的な数値計算を行ったことで、線形計画問題の解法を除いた整数計画問題の求解性能の向上に関わる複数の工夫のうち、順に (1) 切除平面の導入、(2) 前処理の実施そして (3) 分枝変数の選択における工夫がより効果的であることを確認した。なお最後の分枝変数の選択における工夫こそが、本研究で長年提案していた計算手続きにおける要点であった。

本研究において、その有効性について長年主張してき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が、今回の数値計算、特に最新バージョンを用いた数値計算では、その有効性を判断することは出来なかった。これについては、本稿で対象としている整数計画問題の規模程度であれば、近年の数理計画ソフトウェアにみられる求解性能の向上によって、計算量の削減が十分可能な段階に至っているためと判断される。従って、本研究で提案する計算手続きの有効性を検証するための今後の課題としては、さらに大きな規模の問題を対象とした数値実験が必要と考えられる。例えば、これまで計画期間は 2 週間であり、 t は 1 日単位で $T=10$ 日としているものをより長期な計画問題を対象とすることが考えられる。

数理計画法によるモデリングにおいて、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およびチューニングツールの活用も含め、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、FICO Xpress を用いる場合と同様、今後 Gurobi Optimizer などの数理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 23 年度専修大学長期国内研究員の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 [1]、平木 [7]、ジャストインタイム生産システム研究会 [11]、小谷 [16]、黒田他 [18]、門田 [21]、村松 [22]、日本生産管理学会編 [23]、大野 [24]、大野 [25] および大野監修-門田編著 [26] 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 [5], 茨木 [8], [9], 茨木-福島 [10], 今野 [13], 今野-鈴木編 [15], 久保 [17], Beasley (ed.) [42], Carter-Price [45], Martin [60], Nemhauser-Wolsey [62] および Wolsey [70]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 [14], 宮代 [19], 宮代-松井 [20], 柳浦-野々部 [38], Achterberg [39], Chen-Batson-Dang [46], Johnson-Nemhauser-Savelsbergh [52], Linderth-Savelsbergh [55] および Lodi [59]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, 藤江 [6], Atamtürk-Savelsbergh [40], Kallrath (ed.) [53], [54], Linderth-Ralphs [56], Linderth-Lodi [57] および OR/MS Today 誌に掲載される Software Surveys [63] (例えば, Fourer [48])などを参照するとよい。
- (4) 決定変数である初期指示量と「かんぱん方式」における初期かんぱん枚数との対応については, 渡辺 [29], 渡辺-安-平木 [34]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 [29], Watanabe-Hiraki [68]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值 (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 [8], 今野-鈴木編 [15]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理計画モデルの場合, その評価関数値は整数となるため, 実際に行わ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 [21]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 [29]-[33], Watanabe-Hiraki [67], [68]において詳細な数値検証を加えている。
- (11) 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 [32]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 [36]を参照。
- (12) ソルバーの並列処理機能の動向については, 品野-藤江 [27], 品野-Achterberg-藤江 [28], Talbi (ed.) [65] および Xu et al. [71]などを参照するとよい。
- (13) MPS フォーマットについては, 渡辺 [29] および Williams [69]などを参照するとよい。また Linear Programming FAQ のホームページ [58] 上にも MPS フォーマットについての解説がある。

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] 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] 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)。
- [5] 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)。
- [6] 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)。
- [7] 平木秀作, 「自動車の現地生産と部品調達」, 溪水社, 広島 (1996)。
- [8] 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)。
- [9] 茨木俊秀, 「最適化の数学」, 共立出版, 東京 (2011)。
- [10] 茨木俊秀, 福島雅夫, 「最適化の手法」, 共立出版, 東京 (1993)。

- [11] 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).
- [12] 株式会社オクトーバー・スカイ, Gurobi Optimizer ソリューションセミナー 2012 資料 (2012).
- [13] 今野浩, 「整数計画法」, 産業図書, 東京 (1981).
- [14] 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).
- [15] 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).
- [16] 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).
- [17] 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).
- [18] 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).
- [19] 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).
- [20] 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).
- [21] 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).
- [22] 村松林太郎, 「新版生産管理の基礎」, 国元書房, 東京 (1979).
- [23] 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).
- [24] 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).
- [25] 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).
- [26] 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).
- [27] 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).
- [28] 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).
- [29] 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).
- [30] 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).
- [31] 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).
- [32] 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).
- [33] 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理計画モデル-」, 専修経営研究年報, No. 35, pp. 1-28 (2011).
- [34] 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).
- [35] 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).
- [36] 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 所報, No. 68, pp. 1-25 (2008).
- [37] 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 (2) -並列処理機能とチューニング機能の効果-」, 専修大学情報科学研究所 情報科学研究, No. 32, pp. 17-39 (2012).
- [38] 柳浦陸憲, 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).
- [39] Achterberg, T., Constraint Integer Programming, Ph. D. Thesis, Technische Universität Berlin (2007).
- [40] Atamtürk, A. and Savelsbergh, M.W.P., "Integer-Programming Software Systems", *Annals of Operations Research*, Vol. 140, pp. 67-124 (2005).
- [41] 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).
- [42] Beasley, J.E. (ed.), *Advances in Linear and Integer Programming*, Oxford University Press, Oxford (1996).
- [43] Bixby, R.E., "Solving Real-World Linear Programs : A Decade and More of Progress", *Operations Research*, Vol. 50, No. 1, pp. 3-15 (2002).
- [44] 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).

- [45] Carter, M.W. and Price, C.C., *Operations Research : A Practical Introduction*, CRC Press, Boca Raton (2000).
- [46] Chen, D.S., Batson, R.G. and Dang, Y., *Applied Integer Programming : Modeling and Solution*, John Wiley & Sons, New Jersey (2010).
- [47] FICO : Getting Started with Xpress Release 7.3 (2012) ;
Xpress-Mosel Reference Manual Release 3.4 (2012) ;
Xpress-Optimizer Reference Manual Release 23.01 (2012) ;
Xpress-Tuner User Guide (2009).
- [48] Fourer, R., “Linear Programming Software Survey”, *OR/MS Today*, Vol. 38, No. 3, pp. 60-69 (2011).
- [49] Gurobi Optimization : Gurobi Optimizer, <http://www.gurobi.com/>.
- [50] 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).
- [51] ILOG, ILOG CPLEX, 現在は IBM ILOG CPLEX Optimizer.
- [52] 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).
- [53] Kallrath, J. (ed.), *Modeling Languages in Mathematical Optimization*, Kluwer Academic Publishers, Massachusetts (2004).
- [54] Kallrath, J. (ed.), *Algebraic Modeling Systems : Modeling and Solving Real World Optimization Problems*, Springer, Heidelberg (2012).
- [55] 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).
- [56] Linderoth, 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).
- [57] Linderoth, 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).
- [58] Linear Programming FAQ : <http://www.neos-guide.org/NEOS/lp-faq>
- [59] 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).
- [60] Martin, R.K., *Large Scale Linear and Integer Optimization : A Unified Approach*, Kluwer Academic Publishers, Massachusetts (1999).
- [61] 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).
- [62] Nemhauser, G.L. and Wolsey, L.A., *Integer and Combinatorial Optimization*, John Wiley & Sons, New York (1988).
- [63] OR/MS Today Software Surveys : <http://www.orms-today.org/ormsurveys.html>
- [64] Sarker, R.A. and Newton, C.S., *Optimization Modeling : A Practical Approach*, CRC Press, Boca Raton (2008).
- [65] Talbi, E. (ed.), *Parallel Combinatorial Optimization*, John Wiley & Sons, Hoboken, NJ (2006).
- [66] 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).
- [67] 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).
- [68] Watanabe, N. and Hiraki, S., “A Modeling Approach to a JIT-based Ordering System”, *Annals of Operations Research*, Vol. 69, pp. 379-403 (1997).
- [69] Williams, H.P., *Model Building in Mathematical Programming* 4th ed., John Wiley & Sons, Chichester, England (1999).
- [70] Wolsey, L.A., *Integer Programming*, John Wiley & Sons, New York (1998).
- [71] 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).