

# 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 ——

渡 辺 展 男 (専修大学経営学部)

## A Solution to a Integer Programming Problem using Parameter Tuning (2)

—— Computational Experience with Gurobi Optimizer ——

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 to reduce the computational time because the model includes many integral variables. In order to tackle the problem, this paper presents a solution using a mathematical programming software (Gurobi Optimizer) with 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 parameter tuning and the other is to verify the effectiveness of the tuning tool, 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，チューニング，かんばん方式

**Key words** : Integer Programming, Mathematical Programming Software, Parameter 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<sup>(1)</sup>の数理計画モデルを数値実験の対象として、先進的な数理計画ソフトウェアに近年実装されているチューニング機能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<sup>(2)</sup>。

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、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、整数計画法における技法である切除平面、前処理、分枝変数の選択、ヒューリスティクスそしてノードの選択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 [3], [4], [6], [44], [48]。例えば, 代表的な数理計画ソフトウェアである ILOG CPLEX [55] 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 3300 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 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 [3], [4], [47], [48]。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 [53] のバージョン 5.0 にいたる 1991 年から 2012 年までの改善率は, 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 2000 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 [5], [14]。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<sup>(3)</sup>。

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。最後に述べたチューニング機能については, 著者らが従来から使用してきた FICO Xpress [51] に加え, 2013 年春以降にリリースされた Gurobi Optimizer [53] の新たなバージョンにおいても利用できるようになり, 先進的な数理計画ソフトウェアにおいて近年実装が進んでいるものである。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, 整数計画問題を対象としたチューニング機能の概要を述べるとともに, その有効性を検証するために行った数値実験を通して得られた知見を述べることにある。具体的には, (1) 現在, 最高速のソルバーであるといわれている Gurobi Optimizer に新たに実装されたチューニングツールの機能について述べ, (2) 数値実験の結果を通して得られた知見を述べるとともに, 先進的な数理計画ソフトウェアにおける近年の動向などを中心に議論する。

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 節において整数計画問題に対する計算法およびチューニング機能について述べる。4 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 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

[32], [72]。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<sup>(4)</sup>。

## 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 ( $n \in K$ )  
 $I_0^{n(i)}$  : 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 ( $n \in J1$  の場合)

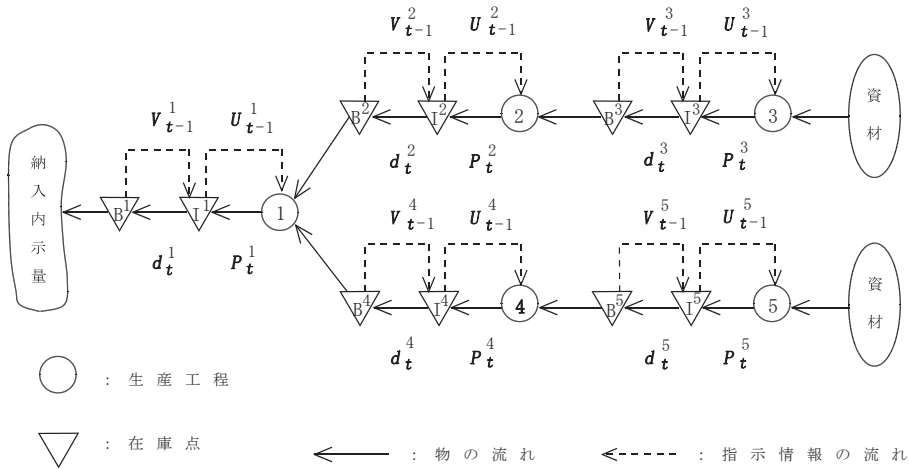


図1 モデルの概念図の例 (N=5)

$P_{j-LP^n}^{n(i)}$  : 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<sup>(5)</sup>。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 [65] 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実際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  $X_i^{n(i)}$ ,  $P_i^{n(i)}$ ,  $d_i^{n(i)}$ ,  $U_0^{n(i)}$ ,  $V_0^{n(i)}$  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  $U_i^{n(i)}$ ,  $V_i^{n(i)}$  および期末在庫量  $I_i^{n(i)}$ ,  $B_i^{n(i)}$  の非負整数性は保証されている。

### 3. 整数計画問題に対する計算法およびチューニング機能

#### 3.1 整数計画問題に対する計算法

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。1 節でも述べたように、整数計画問題を解くためのソルバーの多くは、分枝カット法を実装している。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。

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 [3], [4], [6], [44], [48]。

- (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)

また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。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 i7-3520M であるが、本研究で使用している Gurobi Optimizer は、マルチスレッドであることを検知した後、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<sup>(6)</sup>。

#### 3.2 チューニング機能

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 [45], [52], [54], [63]。

この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については、著者らが従来から使用している FICO Xpress [51] においていち早く実装され、著者らはそのチューニングツールを用いた数値検証を既に実施している [36], [37]。また 2013 年春以降は, Gurobi Optimizer [53] の新たなバージョンにおいてもチューニング作業を支援するツールが利用できるようになり、先進的な数理計画ソフトウェアにおいては、近年その実装が進んでいる。以下では、現在、最高速のソルバーであるといわれている Gurobi Optimizer に新たに実装されたチューニングツールの機能、特に、コマンドライン上で実行されるツールの機能について述べる（これ以降、`grbtune` はチューニングツールを意味しているものとする）。

先進的なソルバーにおいては、ユーザには見えないものも含めると実際は非常に多くの制御パラメータが存在するが、Gurobi Optimizer で新たに提供されているチューニングツール `grbtune` では、57 件のパラメータ設定が対象となっている [29], [53]。チューニングツールの基本的な機能は、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等に関わる制御パラメータの設定を変更しながら、ユーザが定めた実行時間（例えば 3600 秒）、該当の整数計画問題の計算を繰り返し実行するものというものである。その基本的な使い方としては、例えばコマンドライン上で、以下のようなコマンドを発行することでチューニングは開始される。

```
grbtune TuneTimeLimit=3600 exstd.mps (24)
```

この実行によって、対象としている問題の MPS フォーマットの問題ファイル（上記の例では `exstd.mps`）を用いて、まず Gurobi Optimizer におけるパラメータの初期設定（baseline）で解が求められ、その後にはチューニングの手続きが開始される<sup>(7)</sup>。上記の場合、チューニング全体の実行時間が 3600 秒となった時点でチューニングは終了する。あるいは実行途中で手動で打ち切ることとなる。この実行によって、保存されるファイルは以下のとおりである。

`grbtune.log`: チューニング過程のログ

`tune x .prm`: チューニングで良いとされたパラメータの組合せ

`tune x .log`: 上記のパラメータの場合の実行ログ

ここで、`x` とは複数のパラメータの組合せが出力されることを示している。

今回の数値計算で使用した Gurobi Optimizer のバージョンでは、一つのパラメータ設定における 1 回の実行時間は、`TuneTimeLimit` で指定された時間の 10% という設定になっているようである。さらに、一つのパラメータ設定の実行をそれぞれ 2 回行っている。計算中に微小の変動、つまり計算の途中経路を、わざと少しずらすことを行いながら計算の実行を行っている。それぞれの実行は、例えば `seed #1` そして `seed #2` といった表記で示される。また、それぞれの実行において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ている。

## 4. 数値計算例

3 節では、整数計画問題に対する計算法の概要および今回使用したチューニングツールの機能について述べた。今回の数値計算では、現在、最高速のソルバーであるといわれている Gurobi Optimizer を用いた。これに合わせ、本研究においてこれまで解いてきた問題よりも大きな規模の問題を対象とした数値実験を行い、チューニング機能の有効性について検証することとした。つまり、これまで期間  $t$  は 1 日単位で  $T=10$  日としていたものをより長期の計画問題を対象とすることとした。具体的には、 $T=20$  日および  $T=30$  日とした計算を実施した。

### 4.1 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（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) 計画期間については、 $t$  は 1 日単位で  $T=20$  日および  $T=30$  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 ( $T=20$  日の場合)
 
$$D_t^{(1)} = 20, D_t^{(2)} = 15, D_t^{(3)} = 5 \quad (t = 1, 2, 3, 19, 20),$$

$$D_t^{(1)} = 30, D_t^{(2)} = 25, D_t^{(3)} = 5 \quad (t = 4, 5, \dots, 18).$$
  - (b) 納入内示量 ( $T=30$  日の場合)
 
$$D_t^{(1)} = 20, D_t^{(2)} = 15, D_t^{(3)} = 4 \quad (t = 1, 2, 3, 21, 22, 23),$$

$$D_t^{(1)} = 25, D_t^{(2)} = 20, D_t^{(3)} = 5 \quad (t = 11, \dots, 15, 29, 30).$$

$$D_t^{(1)} = 30, D_t^{(2)} = 25, D_t^{(3)} = 5 \quad (t = 4, \dots, 10, 16, \dots, 20, 24, \dots, 28).$$
  - (c) 生産能力
 
$$W_t^n = 480 \text{ 分} \quad (n = 1, 2, \dots, 5; t = 1, 2, \dots, 10).$$
  - (d) 単位量当たり加工時間
 
$$a^{n(i)} = 6 \text{ 分} \quad (i = 1, 2, 3; n = 1, 2),$$

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



(e) 段取り替え時間

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

(f) サブロットの大きさ

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

(g) 初期在庫量

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

(h) 期末目標在庫量

$$SB_t^{n(1)} = 10, SB_t^{n(2)} = 8, SB_t^{n(3)} = 3 \\ SI_t^{n(1)} = 10, SI_t^{n(2)} = 8, SI_t^{n(3)} = 3 \quad (n = 1, 2, \dots, 5; t = 1, 2, \dots, 30).$$

(i) 生産仕掛量

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

なお、部品構成を表す  $e^{sn(i)}$  は全て 1 である。

以上の生産条件の下での実際に解くべき整数計画問題の規模は、(1)  $T=20$  日の場合、制約式が 1306 制約、整数変数は 630 変数であり、(2)  $T=30$  日の場合、制約式が 1956 制約、整数変数は 930 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 i7-3520M 2.90 GHz
(2) RAM	16 GB
(3) オペレーティングシステム (OS)	Windows 7 Professional 64 bit SP1
(4) 数理計画ソフトウェア	Gurobi Optimizer Version 5.5

## 4.2 計算結果と考察

前項 4.1 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  $T=20$  日の場合の数値計算の結果を表 1 に示す。表の見方は次の通り。

- (1) Time : 計算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、今回本稿で計算例として示す 4 件の計算ではいずれもこの場合であった。そのため生成ノードの項に、- と表記している。

表 1 で示されているように初期設定 (baseline) による計算結果から、パラメータの設定によって、最適解へ到達し計算を完了させる時間を 60 秒以内にすることが出来ると判明したため、式 (24) と同様に、実行時間の上限を 3600 秒として grbtune を実行した。3.2 項で述べたように、まずパラメータ

表 1 計算結果 (1)

	Gurobi 5.5 T=20 日 初期設定	Gurobi 5.5 T=20 日 チューニング後
最初の整数解 Time 生成ノード 評価関数値	0 秒 — 575	0 秒 — 575
最良の整数解 Time 生成ノード 評価関数値	5 回目 35 秒 28,594 565	5 回目 7 秒 173 565
最終結果 Time 生成ノード 評価関数値 判 定	61 秒 41,000 565 最適解判定 探索完了	8 秒 225 565 最適解判定 探索完了

の初期設定で 2 回の計算が実施され、seed #1 の計算では 61.42 秒で、seed #2 の計算では 56.42 秒で最適解へ到達し計算が完了した。そして、この計算完了時間の平均値である 58.92 秒を目標値としてチューニングが開始された。今回の計算では 203 件目のパラメータの組合せで計算をしていた 3123.88 秒時点で、手動でチューニングの実行を打ち切った。

チューニングを停止した 3,123.88 秒時点で、grbtune から最後に表示されたログは、以下のとおりである。

Tested 203 parameter sets in 3123.88s

Baseline parameter set : runtime 58.92s

Improved parameter set 1 (runtime 8.32s) :

MIPFocus 2

VarBranch 0

Improved parameter set 2 (runtime 8.81s) :

MIPFocus 2

ここでパラメータ MIPFocus は整数解探索の過程で、実行可能解を得ることを優先させるか、あるいは最適解への到達（最適性の保証）を優先させるかを定めるものである。MIPFocus の初期設定値（Default Value）は 0 で両者のバランスをとった探索戦略を採用しているが、今回のチューニングではこの値を 2、つまり最適性の保証を優先させる方向で整数解探索を行うことが求解を早めることが判明したことになる。なおパラメータ VarBranch は、分枝変数の選択方法を定めるものである。VarBranch の初期設定値は -1 でソルバーが自動設定する。今回のチューニングはこの値を 0、つまり擬コスト（pseudo-cost）に基づいた分枝を行うことが求解を早めることを示している<sup>(8)</sup>。

表 1 で示されているチューニング後の計算とは、パラメータ MIPFocus を 2 として、あらためて Gurobi Optimizer を実行して得られた計算結果を示している。表 1 で示されているように、チューニング機能の有効性は明らかである。

(1) 最適解への到達については、チューニング後は、初期設定と比べ、1/5 の計算時間（7 秒と 35 秒）

と 1/165 以下の生成ノード数 (173 と 28,594) で達成している。

- (2) 最適判定を行い解の探索を完了することについては、チューニング後は、初期設定と比べ、1/7 以下の計算時間 (8 秒と 61 秒) と 1/182 以下の生成ノード数 (225 と 41,000) で達成している。

次に  $T=30$  日の場合の数値計算の結果を表 2 に示す。表の見方は表 1 と同じである。

$T=30$  日の場合、現在、最高速のソルバーであるといわれている Gurobi Optimizer を用いても初期設定では、計算を完了させることは出来なかった。また表 2 で示されているチューニング後の計算とは、 $T=20$  日の場合、チューニングで有効とされたパラメータ MIPFocus を 2 として、Gurobi Optimizer を実行して得られた計算結果を示している。

表 2 で示されているように、 $T=30$  日の場合においても、チューニング機能の有効性は明らかである。

- (1) 最適解への到達については、チューニング後は、初期設定と比べ、1/3.7 以下の計算時間 (282 秒と 1,058 秒) と 1/132 以下の生成ノード数 (2,690 と 357,071) で達成している。
- (2) 最適判定を行い解の探索を完了することについては、計算時間 2058 秒、生成ノード 1,411,209 の時点で達成しているチューニング後の計算に対して、初期設定では計算時間の上限である 3600 秒の時点で最適判定を行うことが出来ず計算が打ち切りとなっている。なおこの時点での生成ノード数は 2,397,539 に達している。

上述のように、今回の数値計算によってチューニングツールの有効性を検証することが出来たが、なお課題が残っていることも判明した。

表 2 で示されているように、初期設定においては、最良の整数解が 1,058 秒で得られているため、 $T=30$  日の場合においては、実行時間の上限を 18,000 秒として grbtune を実行してみた。これまで述べてきたように、まずパラメータの初期設定で 2 回の計算が実施されるが、seed #1 の計算、seed #2 の計算ともに、上限設定値の 10% である 1,800 秒では計算は完了出来ず、上限値と下限値とのギャップ(gap)がともに 0.18% で計算は打ち切られた。そして、このギャップの値 0.18% を目標値としてチューニングが開始された。このチューニングの結果であるが、6 件目のパラメータの組合せで計算をしていた時点で、実行の上限値である 18,000 秒となり実行が打ち切られた。

チューニングを停止した 18,000.05 秒時点で、grbtune から最後に表示されたログは、以下のとおりである。

表 2 計算結果 (2)

	Gurobi 5.5 $T=30$ 日 初期設定	Gurobi 5.5 $T=30$ 日 チューニング後
最初の整数解 Time 生成ノード 評価関数値	3 秒 — 3,000	19 秒 — 1,595
最良の整数解 Time 生成ノード 評価関数値	18 回目 1,058 秒 357,071 560	10 回目 282 秒 2,690 560
最終結果 Time 生成ノード 評価関数値 判 定	3,600 秒 2,397,539 560 計算打ち切り	2,058 秒 1,411,209 560 最適解判定 探索完了

Tested 6 parameter sets in 18000.05s

Baseline parameter set : MIP gap 0.18%

Unable to improve on baseline parameter set.

つまり、18,000 秒のチューニング時間内では、初期設定 (baseline) よりも求解性能に優れたと判断されるパラメータの組合せを探ることができていない。従って表 2 で示したように、 $T=30$  日の場合においても、 $T=20$  日の場合のチューニングで有効とされたパラメータ MIPFocus の設定値 2 を用いた計算を実行してみた。

Gurobi Optimizer を用いてもパラメータの初期設定では、最適解へ到達した後、最適判定を行い計算を完了させることが出来ない規模の問題を解いているという点も確かにあるが、2013 年春リリースの新たなバージョンにおいて初めて実装したチューニングツールの機能にも改善の余地があるものと考えられる。例えば、(1) 1 回毎のチューニング実行時間をユーザが指定できる機能、(2) この 1 回毎の実行時間の他、分枝変数の選択の優先順位情報の取り込み、繰り返し計算される線形計画問題で採用する解法の設定、並列計算を行うか否か等も含めチューニングの実行そのものに関係する種々の設定を行うことができるインターフェースの提供などである。

#### 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 1 のチューニング後の計算において得られている値である。

$$\begin{aligned} U_0^{1(1)} &= 31, U_0^{1(2)} = 26, U_0^{1(3)} = 3; & V_0^{1(1)} &= 26, V_0^{1(2)} = 21, V_0^{1(3)} = 3; \\ U_0^{2(1)} &= 27, U_0^{2(2)} = 32, U_0^{2(3)} = 11; & V_0^{2(1)} &= 26, V_0^{2(2)} = 21, V_0^{2(3)} = 8; \\ U_0^{3(1)} &= 34, U_0^{3(2)} = 26, U_0^{3(3)} = 10; & V_0^{3(1)} &= 26, V_0^{3(2)} = 26, V_0^{3(3)} = 8; \\ U_0^{4(1)} &= 26, U_0^{4(2)} = 21, U_0^{4(3)} = 3; & V_0^{4(1)} &= 26, V_0^{4(2)} = 21, V_0^{4(3)} = 3; \\ U_0^{5(1)} &= 26, U_0^{5(2)} = 21, U_0^{5(3)} = 3; & V_0^{5(1)} &= 26, V_0^{5(2)} = 21, V_0^{5(3)} = 3. \end{aligned}$$

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。

## 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、現在、最高速のソルバーであるといわれている Gurobi Optimizer に新たに実装されたチューニングツールの機能について述べ、これを用いて行った数値実験の結果を通して得られた知見を述べた。本稿では求解性能についての比較検討を行っていないが、著者らが従来から使用してきた FICO Xpress と比べ、整数解を求めることについては Gurobi Optimizer の求解の速さに驚いた。このソルバーを用いることで著者らが長年願っていたより大きな規模の問題、例えば、月次レベルの生産計画問題に対応した問題を対象とした数値実験が可能となった。具体的には、 $T=20$  日および  $T=30$  日とした計算を実施し、今回の数値計算によって、本稿のねらいであったチューニング機能の有効性を検証することが出来た。その一方で、 $T=30$  日とした計算においては、Gurobi Optimizer を用いた場合においても

多くの計算時間が必要であり、またチューニング機能についても改善の余地が考えられることが判明した。

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、これまで本研究で長年使用してきた FICO Xpress が提供するシステムそして本稿で示した Gurobi Optimizer も含め数理計画ソフトウェアを用いる場合において、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 24 年度専修大学研究助成・個別研究・研究課題「最適化ツールを用いた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 [9], ジャストインタイム生産システム研究会 [13], 小谷 [18], 黒田他 [20], 門田 [23], 村松 [24], 日本生産管理学会編 [25], 大野 [26], 大野 [27] および大野監修・門田編著 [28]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 [6], 茨木 [10], [11], 茨木・福島 [12], 今野 [15], 今野・鈴木編 [17], 久保 [19], Beasley (ed.) [46], Carter - Price [49], Martin [64], Nemhauser - Wolsey [66] および Wolsey [74]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 [16], 宮代 [21], 宮代・松井 [22], 柳浦・野々部 [42], Achterberg [43], Chen - Batson - Dang [50], Johnson - Nemhauser - Savelsbergh [56], Linderth - Savelsbergh [59] および Lodi [63]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, Berthold et al. [2], 藤江 [7], Atamtürk - Savelsbergh [44], Kallrath (ed.) [57], [58], Linderth - Ralphs [60], Linderth - Lodi [61] および OR/MS Today 誌に掲載される Software Surveys [67] (例えば, Fourer [52])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 [32], 渡辺・安・平木 [38]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 [32], Watanabe - Hiraki [72]を参照。
- (6) ソルバーの並列処理機能の動向については, 品野・藤江 [30], 品野・Achterberg - 藤江 [31], Talbi (ed.) [69] および Xu et al. [75]などを参照するとよい。
- (7) MPS フォーマットについては, 渡辺 [32] および Williams [73]などを参照するとよい。また Linear Programming FAQ のホームページ [62] 上にも MPS フォーマットについての解説がある。
- (8) 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 [10], 今野・鈴木編 [17]などを参照するとよい。

## 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] 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).
- [3] 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).
  - [4] Bixby, R.E.: “Progress in Optimization & Gurobi Optimizer”, Gurobi Optimizer 販売記念特別講演会資料 (2010).
  - [5] 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).
  - [6] 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).
  - [7] 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).
  - [8] 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).
  - [9] 平木秀作: 「自動車の現地生産と部品調達」, 溪水社, 広島 (1996).
  - [10] 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).
  - [11] 茨木俊秀: 「最適化の数学」, 共立出版, 東京 (2011).
  - [12] 茨木俊秀, 福島雅夫: 「最適化の手法」, 共立出版, 東京 (1993).
  - [13] 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).
  - [14] 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).
  - [15] 今野浩: 「整数計画法」, 産業図書, 東京 (1981).
  - [16] 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).
  - [17] 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).
  - [18] 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).
  - [19] 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).
  - [20] 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).
  - [21] 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).
  - [22] 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).
  - [23] 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).
  - [24] 村松林太郎: 「新版生産管理の基礎」, 国元書房, 東京 (1979).
  - [25] 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).
  - [26] 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).
  - [27] 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).
  - [28] 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).
  - [29] 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).
  - [30] 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).
  - [31] 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).
  - [32] 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).
  - [33] 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).
  - [34] 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).
  - [35] 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).
  - [36] 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).
  - [37] 渡辺展男: “数理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).
  - [38] 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).

- [39] 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).
- [40] 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).
- [41] 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).
- [42] 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).
- [43] Achterberg, T.: Constraint Integer Programming, Ph. D. Thesis, Technische Universität Berlin (2007).
- [44] Atamtürk, A. and Savelsbergh, M.W.P.: “Integer-Programming Software Systems”, *Annals of Operations Research*, Vol. 140, pp. 67-124 (2005).
- [45] 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).
- [46] Beasley, J.E. (ed.): *Advances in Linear and Integer Programming*, Oxford University Press, Oxford (1996).
- [47] Bixby, R.E.: “Solving Real-World Linear Programs: A Decade and More of Progress”, *Operations Research*, Vol. 50, No. 1, pp. 3-15 (2002).
- [48] 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).
- [49] Carter, M.W. and Price, C.C.: *Operations Research: A Practical Introduction*, CRC Press, Boca Raton (2000).
- [50] Chen, D.S., Batson, R.G. and Dang, Y.: *Applied Integer Programming: Modeling and Solution*, John Wiley & Sons, New Jersey (2010).
- [51] FICO: Getting Started with Xpress Release 7.5 (2013);  
Xpress-Mosel Reference Manual Release 3.4 (2013);  
Xpress-Optimizer Reference Manual Release 24.01 (2013);  
Xpress-Tuner User Guide Release 1.1 (2009).
- [52] Fourer, R.: “Linear Programming Software Survey”, *OR/MS Today*, Vol. 40, No. 3, pp. 40-53 (2013).
- [53] Gurobi Optimization:  
Gurobi Optimizer Reference Manual Version 5.5 (2013);  
Webinar: “Recent Improvements in the Gurobi Optimizer” (April, 2013).
- [54] 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).
- [55] ILOG: ILOG CPLEX, 現在は IBM ILOG CPLEX Optimizer.
- [56] 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).
- [57] Kallrath, J. (ed.): *Modeling Languages in Mathematical Optimization*, Kluwer Academic Publishers, Massachusetts (2004).
- [58] Kallrath, J. (ed.): *Algebraic Modeling Systems: Modeling and Solving Real World Optimization Problems*, Springer, Heidelberg (2012).
- [59] 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).
- [60] 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).
- [61] 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).
- [62] Linear Programming FAQ: <http://www.neos-guide.org/NEOS/lp-faq>
- [63] 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).
- [64] Martin, R.K.: *Large Scale Linear and Integer Optimization: A Unified Approach*, Kluwer Academic Publishers, Massachusetts (1999).
- [65] 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).
- [66] Nemhauser, G.L. and Wolsey, L.A. : Integer and Combinatorial Optimization, John Wiley & Sons, New York (1988).
  - [67] OR/MS Today Software Surveys : <http://www.orms-today.org/ormssurveys.html>
  - [68] Sarker, R.A. and Newton, C.S. : Optimization Modeling : A Practical Approach, CRC Press, Boca Raton (2008).
  - [69] Talbi, E. (ed.) : Parallel Combinatorial Optimization, John Wiley & Sons, Hoboken, NJ (2006).
  - [70] 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).
  - [71] 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).
  - [72] Watanabe, N. and Hiraki, S. : "A Modeling Approach to a JIT-based Ordering System", *Annals of Operations Research*, Vol. 69, pp. 379-403 (1997).
  - [73] Williams, H.P. : Model Building in Mathematical Programming 5th ed., John Wiley & Sons, Chichester, England (2013).
  - [74] Wolsey, L.A. : Integer Programming, John Wiley & Sons, New York (1998).
  - [75] 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).