

法政大学 理工学部 卒業論文

# Max-plus 線形システムでの 1 段階法と 2 段階法とによる最適化

経営システム工学科

学籍番号：10X4058

氏名：豊田一裕

指導教員：五島洋行 教授

学科名	経営システム工	学籍番号	10X4058
申請者氏名	豊田 一裕		
指導教員 氏 名	五島 洋行		

## 論 文 要 旨

論文題目	Max-plus 線形システムでの 1 段階法と 2 段階法とによる最適化
------	--

本論文では、max-plus 代数を使用して未来のシステムの状態を数式化し、そのシステムの作業時間と部品の到着時間についての最適化問題を、線形計画法を用いて解く。この最適化問題を解く際、従来は最初に作業時間についての最適化問題を解いた後に、部品の到着時間の最適化問題を解く、というように 2 段階に分けて最適化を行ってきた。しかし 2 段階に分けることで最適解にロスが生まれる可能性がある。このロスを解消するために、作業時間と部品の到着時間を同時に解く 1 段階法を提案し、2 段階法との比較を行う。

本研究で用いる max-plus 線形システムとは、max 演算と plus 演算を基本演算とする max-plus 代数を用いてモデル化したシステムのことである。Max-plus 線形システムを用いることで、多入力多出力システムでのシステムの内部の状態を簡単な線形表記で表すことができ、生産システムのスケジュールを組むことが可能になる。

多入力多出力のサンプルシステムを用いて数値実験を行った結果、従来法では、同じ部品を時間差で違う作業で使う場合に、部品をストックしておく時間が生まれていた。本論文で提案した 1 段階法では、部品をストックしておく時間を短くし、時間のロスを抑えることに成功している。また 1 段階法で得られた目的関数値は、2 段階法で得られた目的関数値よりも優れていた。Max-plus 線形システムにおいて、作業時間と外部からの部品の到着時刻に関する最適化を同時に行う 1 段階法は有益な方法と言える。

## 目次

第 1 章	序論.....	1
1. 1	研究背景 .....	1
1. 2	研究目的 .....	2
1. 3	論文の構成.....	2
第 2 章	Max-plus 線形システムによるスケジューリング .....	3
第 3 章	Max-plus 代数.....	5
3. 1	Max-plus 代数の定義.....	5
3. 2	Max-plus 線形システムとは.....	6
3. 3	Max-plus 線形システムでのモデル予測制御 .....	9
第 4 章	提案手法 .....	13
4. 1	線形計画法への適用.....	13
4. 2	2 段階法.....	14
4. 3	1 段階法.....	15
第 5 章	数値実験 .....	16
5. 1	様々なシステムでの数値実験 .....	16
5. 2	従来法との性能の比較 .....	21
5. 3	評価 .....	23
第 6 章	結論.....	25
参考文献	.....	26

# 第 1 章 序論

## 1. 1 研究背景

本研究では、生産システムなどの、有向非サイクルグラフで表せる、多入力多出力の反復的に行われるシステムに焦点を当て、考察を行う。

同じ構造のシステムを繰り返し実行する離散事象システムと呼ばれる生産システムに対して、システムのスケジューリングやモデル化を行う手法の一つに、Max-plus 代数と呼ばれる特別代数系を用いる手法がある。Max-plus 代数を用いることで、システム内の振る舞いを簡単な線形の式で表すことができ、そのようなシステムを Max-plus 線形システムと呼ぶ。

Max-plus 線形システムを利用する利点としては、システムのモデル化により入力出力間の関係を線形表記で与えることができ、未来のシステムの状態を予測し、望ましい出力結果を得るためには、どのような入力を設定すればよいかを設計できる点である。このようにシステムの状態を予測することを、モデル予測制御 (Model Predictive Control; MPC) と呼ぶ。このような Max-plus 線形システムによるモデル予測制御は生産システムのほかにも、待ち行列システムや鉄道システムでも応用が試みられている。Max-plus 代数、Max-plus 線形システムおよび、MPC に関する詳しい定義や定式化については第 3 章で述べる。

線形計画法は生産計画分野へ様々な応用がされており、生産スケジューリング、人員スケジューリング、輸送計画問題などが代表例である。この線形計画問題を解く手法として、シンプレックス法 (単体法) や内点法がある。その中でもシンプレックス法は、George B. Dantzig によって 1940 年頃に開発された [1]。この方法は、問題が実行可能であるという前提の下で、実行可能解からスタートし、目的関数値が改善されるような実行可能基底解を求める手法である。シンプレックス法は様々なソフトウェアで実行され、パーソナルコンピュータ上でスプレッドシートの形で利用可能である。また、非退化の場合、有限回で解が収束することが示されており、実用的には指数時間で最適解が得られることが知られている。

これらを組み合わせることにより、未来のシステムの状態を予測し、コストが少なくより効率的な生産システムの構築の実現を目指す。

## 1. 2 研究目的

一般的に企業の生産の場では、生産時間を短縮することや、外部からの部品の納期を早めることでコストがかかると考えられる。本研究では、システムの納期に対して遅れを最小にしつつ、作業にかかる時間を長くとり、外部からの入力をできるだけ遅らせるようなスケジューリング手法を扱う。従来法は、はじめに作業にかかる最適な時間を求めた後に、部品の最適な到着時間を求めるというように、2段階に分けて最適化問題を解いてきた。しかし、2段階に分けて最適化問題を解くことでどこかでロスが生まれる可能性がある。この最適化問題を2段階に分けず1段階で解くことで、より最適な解が得られるのではないかと考える。

この最適化問題を  $\max\text{-plus}$  線形システムで適用するために、第一に  $\max\text{-plus}$  代数を用いてシステムの内部の状態を定式化し、第二にシステムの状態の式に基づいて未来の状態を予測し、第三に線形計画問題を解く。

## 1. 3 論文の構成

本論文は全6章で構成されている。

第2章では、 $\max\text{-plus}$  代数、 $\max\text{-plus}$  線形システム、およびそれらを用いたモデル予測制御を行った研究について紹介する。

第3章では、 $\max\text{-plus}$  代数についての詳しい定義や、システムの振る舞いを線形で表現する方法について述べる。

第4章では、 $\max\text{-plus}$  線形システムの最適化を行うための定式化を行う。また1段階法について提案する。

第5章では、さまざまなシステムでの数値実験を行い、1段階法と2段階法との性能についての評価を行う。

第6章では、本論文のまとめについて述べる。

## 第 2 章 Max-plus 線形システムによる スケジューリング

生産システムなどの離散事象システムをモデリングする手法の一つに、max-plus 代数と呼ばれる代数系を用いる方法がある。Max-plus 代数とは、max 演算を + 演算、通常の + 演算を乗算と定義する代数系のことであり、主なメリットは、システムの内部の状態を簡単な線形の式で表すことができる点である。Max-plus 代数は B. A. Carre [2]によりアイデアが出され、R. A. Cuninghame-Green [3]や G. Cohen & D. Dubois [4]らにより一つの理論体系となった。

Max-plus 線形システムとは、max-plus 代数を用いてモデル化されるシステムモデルのことである。Max-plus 線形システムは様々な手法でスケジューリングされている。笠原 [5]は、max-plus 線形システムをクリティカルチェーン法に適用し、納期短縮と納期遅れを防止するスケジューリング手法を提案した。また造船ラインのスケジューリング [6]などにも応用されており、今後も様々な場でのスケジューリングへの適用に期待されている。

また max-plus 線形システムでは、離散事象システムの内部の振る舞いを、現代制御理論での状態空間法に似た形で表すことができる。これまで制御理論においては時間変数を独立変数にとっているが、max-plus 線形システムはイベントカウンターと呼ばれる事象の発生回数を表す独立変数を用いる。事象の発生回数を独立変数にとることで、事象駆動システムに対してもモデル予測制御 (MPC) の設計が可能となった。Max-plus 代数、max-plus 線形システムおよび、モデル予測制御に関しての詳しい定義や定式化については第 3 章で述べる。

Max-plus 線形システムを用いた離散事象システムのモデリングとモデル予測制御に関する研究は、S. Masuda と H. Goto ら [7] や B. De. Schutter と T. Boom ら [8] [9]によって盛んに研究報告がされてきた。その中でも、Goto2012 [10]では、容量制約を持つ、反復的な離散事象システムでのモデル予測制御を行っている。文献 [8]によって提案された、制約条件を線形制約条件に緩和し、最適解の導出を簡単化することを用いて、最適な作業時間と外部からの入力時刻を求めるアプローチを採る検討を行う。この最適化問題を解く際、最初に生産にかかる時間に関しての最適化を行った後、外部からの部品の入力時刻に関する最適化を行っている。Max-plus 線形システムでのモデル予測を行った予測式から導出された制約条件には、max-plus 代数の演算子が含まれる。そのため max-plus 代数が含まれる最適化問題を、いかに効率よく、またいか

に性能よく解くかが課題となっている [7]. そこで本研究は, 従来は作業時間と外部からの部品の入力時刻との最適化を 2 段階に分けて解いてきたものを, 同時に解く 1 段階法を提案する.

# 第 3 章 Max-plus 代数

## 3. 1 Max-plus 代数の定義

Max-plus代数とはmax演算と+演算を基本演算とする代数系のことである。主に離散事象システムにおいて、事象の発生回数によって、システム内の状態がどのように変化していくかを記述することができる [11]。このような利点により、特にスケジューリングを目的とする生産システムのモデリングや、モデル予測制御 (MPC) の分野で様々な応用が試みられている。

Max-plus 代数における加算と乗算は以下のように定義する。

$$x \oplus y = \max\{x, y\} \quad (1)$$

$$x \otimes y = x + y \quad (2)$$

また、 $\oplus$ の単位元を $\varepsilon = -\infty$ 、 $\otimes$ の単位元を $e=0$ と定める。これにより以下の等式が成り立つ。

$$x \oplus \varepsilon = \varepsilon \oplus x = x \quad (3)$$

$$x \otimes e = e \otimes x = x \quad (4)$$

通常の計算で扱われる演算子での乗法と除法が、加法と減法より計算の優先度が高いのと同様に、max-plus 代数で扱われる演算子の優先度は、plus 演算子 $\otimes$ の方が max 演算子である $\oplus$ より計算の優先順位が高い。

以上のように定義した max-plus 代数は、以下の法則が成り立つことが知られている。

・交換法則

$$x \oplus y = y \oplus x \quad (5)$$

$$x \otimes y = y \otimes x \quad (6)$$

・結合法則

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z \quad (6)$$

$$x \otimes (y \otimes z) = (x \otimes y) \otimes z \quad (7)$$

・分配法則



$$x \oplus (y \otimes z) = x \otimes y \oplus x \otimes z \quad (8)$$

複数の数 $x_k$ に対する演算子は以下のように表記する.

$$\bigoplus_{k=1}^n x_k = x_1 \oplus x_2 \oplus \cdots \oplus x_n = \max\{x_1, x_2, \dots, x_n\} \quad (9)$$

$$\bigotimes_{k=1}^n x_k = x_1 \otimes x_2 \otimes \cdots \otimes x_n = x_1 + x_2 + \cdots + x_n \quad (10)$$

また **max-plus** 代数では, 行列やベクトルの線形計算においても, 通常の演算子と同様の計算法で計算することができる.  $m$  行  $n$  列の行列  $A$  と,  $n$  行  $l$  列の行列  $B$  を用いると,

$$\begin{aligned} [A \oplus B]_{ij} &= \bigoplus_{k=1}^n [A]_{ik} \otimes [B]_{kj} \\ &= ([A]_{i1} \otimes [B]_{1j}) \oplus ([A]_{i2} \otimes [B]_{2j}) \oplus \cdots \oplus ([A]_{in} \otimes [B]_{nj}) \end{aligned} \quad (11)$$

と表現できる. これは通常の演算子の加法と乗法を, それぞれ $\oplus$ と $\otimes$ に置き換えたものに相当する. **Max-plus** 代数を用いても, 線形演算は通常の演算子と同様に定義できるが,  $\oplus$ の逆元が存在しないため, 逆行列は通常と同じように定義できない.

## 3. 2 Max-plus 線形システムとは

**Max-plus** 線形システムとは, **max-plus** 代数を用いてモデル化されるシステムモデルのことであり, 主に事象駆動システムの振る舞いをモデル化することに適している. 事象駆動システムとは, 人間が介在する人工システムで, 時間の経過だけでなく, 事象の生起によりシステムの内部の状態に変化が起こるシステムのことを指す. 例えば, 企業の製品の生産は, 顧客の依頼や発注という事象により活動がはじまり, 生産工程内部の状態に変化が起こる.

事象駆動システムには以下のような事象生起条件を持つ.

- 各作業を1回だけ通過する.
- 各作業は, 1回前の作業が終了するまで次の作業を開始できない.
- 先行作業が終了するまで, 後続作業は開始できない.
- 外部からの部品の入力がある作業は, 部品の入力があるまで作業を開始できない.

この条件を, 先行作業の完了時間  $X_1, X_2, \dots, X_n$ , 後続作業の開始時刻  $Y$  として式で表すと以下の式になる.

$$Y = \max(X_1, X_2, \dots, X_n) = \bigoplus_{k=1}^n X_k \quad (12)$$

また一般的なシステムや生産の場では, 同じ条件下で何度も実行が繰り返される. そのため, システム内の動的な振る舞いを予測するには, 何度も事象が生起することを想定する必要がある. そこで, 初期状態からの事象発生回数を表す, イベントカウンターと呼ばれる独立変数  $k$  を導入する.

これらのことを考慮して, **max-plus** を用いてシステムの振る舞いを表す. ここで, 数式表現するのに必要な変数を以下のように設定する.

- $x_i(k)$ : 作業  $i$  での作業完了時刻.
- $d_i(k)$ : 作業  $i$  で要する作業時間.
- $u_i(k)$ : 部品  $i$  を作業に投入する時刻.
- $y(k)$ : 完成品を送り出す時間.
- $r(k)$ : 完成品の納期.
- $\mu(k)$ : 納期に対する遅れ.

これらをベクトル表記すると, 以下の式になる.

$$\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T \quad (13)$$

$$\mathbf{y}(k) = y(k) \quad (14)$$

$$\mathbf{u}(k) = [u_1(k), u_2(k), \dots, u_n(k)]^T \quad (15)$$

これらのことを考慮して, システムを記述したものが **max-plus** 線形システムで, 以下の式のように状態方程式に似た形式で与えられる.

$$\mathbf{x}(k+1) = \mathbf{A}_k \otimes \mathbf{x}(k) \oplus \mathbf{B}_k \otimes \mathbf{u}(k+1) \quad (16)$$

$$y(k) = C \otimes x(k+1) \quad (17)$$

$A_k, B_k, C$  はそれぞれシステム行列, 入力行列, 出力行列と呼ばれ, 詳細は次節で述べる.

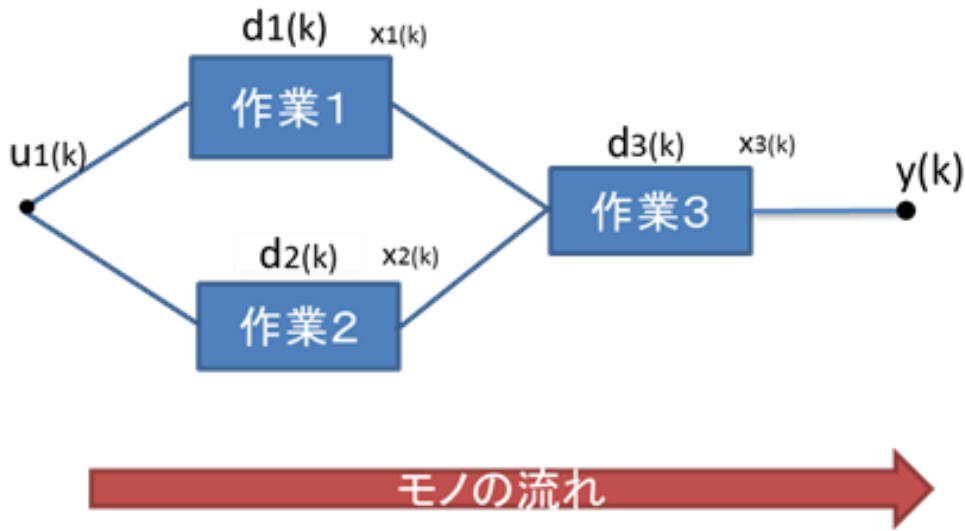


図 1 1 入力 1 出力システムの例

ここで max-plus 線形システムの具体例をあげる. 図 1 に示した 1 入力 1 出力の生産システムでの max-plus 代数での線形表記について考える. この生産システムは三つの作業工程と, 一つの外部からの部品の入力, 一つの完成品の出力から構成されている. 作業 1, 2 に外部から同じ部品を投入した後, 作業 3 にて作業 1, 2 でできた部品を加工し 1 つの製品にし出荷する. 各作業にかかる時間はそれぞれ  $d_1, d_2, d_3$  で, 外部からの部品の到着時刻は  $u_1$ , 完成品の出荷時刻を  $y$  で表す. 各作業の準備時間や作業間の運搬時間はかからないものとし 0 と考える.

$x_1$  と  $x_2$  のそれぞれの  $k+1$  番目の作業の終了時刻は,  $k$  番目の作業と  $k+1$  の作業時間の和と,  $k+1$  番目の部品の到着時刻と  $k+1$  番目の作業時間の和の最大値である. この関係を max-plus 代数を使った表記で表すと以下の式になる.

$$x_1(k+1) = x_1(k) \otimes d_1(k+1) \oplus u_1(k+1) \otimes d_1(k+1) \quad (18)$$

$$x_2(k+1) = x_2(k) \otimes d_2(k+1) \oplus u_1(k+1) \otimes d_2(k+1) \quad (19)$$

また,  $x_3$  の  $k+1$  番目の作業終了時刻は,  $x_1$  と  $x_2$  のそれぞれの  $k+1$  番目の作業の終了時刻

に作業 3 の作業時間を足したものと,  $x_3$  の  $k$  番目の作業終了時刻に作業 3 の作業時間を足したものの最大値である. この関係を max-plus 代数を用いて表すと以下の式になる.

$$\begin{aligned} x_3(k+1) = & x_1(k) \otimes d_1(k+1) \otimes d_3(k+1) \oplus x_2 \otimes d_2(k+1) \otimes d_3(k+1) \\ & \oplus x_3 \otimes d_3(k+1) \oplus u_1 \otimes d_1(k+1) \otimes d_3(k+1) \\ & \oplus u_1 \otimes d_1(k+1) \otimes d_3(k+1) \end{aligned} \quad (20)$$

そして, 外部への完成品の出力を持つ作業 3 の  $k+1$  番目の作業終了時刻が, そのままこの生産システム全体の  $k+1$  番目の終了時刻となり, この関係を式に表すと,

$$y(k+1) = [\varepsilon \quad \varepsilon \quad e] \otimes x(k+1) \quad (21)$$

となる. 式(19)~(22)で与えられた式を, 式(17), (18)の形に線形表記すると,

$$x(k+1) = \begin{bmatrix} d_1(k+1) & \varepsilon & \varepsilon \\ \varepsilon & d_2(k+1) & \varepsilon \\ d_1(k+1) \otimes d_3(k+1) & d_2(k+1) \otimes d_3(k+1) & d_3(k+1) \end{bmatrix} \otimes x(k) \quad (22)$$

$$\oplus \begin{bmatrix} d_1(k+1) & & \\ & d_2(k+1) & \\ d_1(k+1) \otimes d_3(k+1) \oplus d_1(k+1) \otimes d_3(k+1) & & \end{bmatrix} \otimes u_1(k+1)$$

$$y(k+1) = [\varepsilon \quad \varepsilon \quad e] \otimes x(k+1) \quad (23)$$

となる.

### 3. 3 Max-plus 線形システムでのモデル予測制御

モデル予測制御 (MPC) とは,

1. 制御する対象のモデルを用いた出力予測式の導出
2. 予測式に基づく最適なシステム入力の決定

によって構成される制御手法のことである [11]. モデル予測制御のメリットは, 望ましい結果を得るためにはどのような入力を設定すればよいか, システムモデルを用いて設計できる点である. モデル予測制御は時間軸を独立変数として持つ時間駆動システムのみで扱われてきたが, max-plus 線形システムを用いることで, 事象の発生回数を独立変数として持つ事象駆動システムに対してもモデル予測制御の設計が可能となった.

式(17)は  $k$  回目の状態から  $k+1$  回目の状態を予測する予測式である。さらに、式(17)の  $k$  を  $k+1$  に、 $k+1$  を  $k+2$  にそれぞれ置き換えると、 $k+2$  回目の予測式を求めることができる。

$$\mathbf{x}(k+2) = \mathbf{A}_{k+1} \otimes \mathbf{x}(k+1) \oplus \mathbf{B}_{k+1} \otimes \mathbf{u}(k+2) \quad (24)$$

この式に  $\mathbf{x}(k+1)$  が含まれているので、式(17)を代入して  $\mathbf{x}(k+1)$  を消去し、まとめると、

$$\mathbf{x}(k+2) = \mathbf{A}_{k+1} \otimes \mathbf{A}_k \otimes \mathbf{x}(k) \oplus \mathbf{A}_{k+1} \otimes \mathbf{B}_k \otimes \mathbf{u}(k+1) \oplus \mathbf{B}_{k+1} \otimes \mathbf{u}(k+2) \quad (25)$$

となり、 $k$  回目の状態から  $k+2$  回目の状態を予測した予測式であることが分かる。同様のことを  $k+3$ ,  $k+4$ ,  $\dots$ ,  $k+N$  と繰り返すと、ステップ先の状態の予測式を導き出すことができる。そのときの予測式は以下のように記述できる。

$$\left[ \begin{array}{l} \mathbf{x}(k+1) = \mathbf{A}_k \otimes \mathbf{x}(k) \oplus \mathbf{B}_k \otimes \mathbf{u}(k+1) \\ \mathbf{x}(k+2) = \mathbf{A}_{k+1} \otimes \mathbf{A}_k \otimes \mathbf{x}(k) \oplus \mathbf{A}_{k+1} \otimes \mathbf{B}_k \otimes \mathbf{u}(k+1) \oplus \mathbf{B}_{k+1} \otimes \mathbf{u}(k+2) \\ \vdots \\ \mathbf{x}(k+N) = \mathbf{A}_{k+N-1} \otimes \dots \otimes \mathbf{A}_k \otimes \mathbf{x}(k) \\ \quad \oplus \mathbf{A}_{k+N-1} \otimes \dots \otimes \mathbf{A}_{k+1} \otimes \mathbf{B}_k \otimes \mathbf{u}(k+1) \\ \quad \oplus \mathbf{A}_{k+N-1} \otimes \dots \otimes \mathbf{A}_{k+2} \otimes \mathbf{B}_{k+1} \otimes \mathbf{u}(k+2) \\ \quad \oplus \dots \oplus \mathbf{B}_{k+N-1} \otimes \mathbf{u}(k+N) \end{array} \right. \quad (26)$$

この予測式の両辺に出力行列  $\mathbf{C}$  を掛けることで、出力予測式を導出できる。その式は、次のように表せる

$$\left[ \begin{array}{l}
\mathbf{y}(k+1) = \mathbf{C} \otimes \mathbf{A}_k \otimes \mathbf{x}(k) \oplus \mathbf{C} \otimes \mathbf{B}_k \otimes \mathbf{u}(k+1) \\
\mathbf{y}(k+2) = \mathbf{C} \otimes \mathbf{A}_{k+1} \otimes \mathbf{A}_k \otimes \mathbf{x}(k) \\
\quad \oplus \mathbf{C} \otimes \mathbf{A}_{k+1} \otimes \mathbf{B}_k \otimes \mathbf{u}(k+1) \oplus \mathbf{C} \otimes \mathbf{B}_{k+1} \otimes \mathbf{u}(k+2) \\
\vdots \\
\mathbf{y}(k+N) = \mathbf{C} \otimes \mathbf{A}_{k+N-1} \otimes \cdots \otimes \mathbf{A}_k \otimes \mathbf{x}(k) \\
\quad \oplus \mathbf{C} \otimes \mathbf{A}_{k+N-1} \otimes \cdots \otimes \mathbf{A}_{k+1} \otimes \mathbf{B}_k \otimes \mathbf{u}(k+1) \\
\quad \oplus \mathbf{C} \otimes \mathbf{A}_{k+N-1} \otimes \cdots \otimes \mathbf{A}_{k+2} \otimes \mathbf{B}_{k+1} \otimes \mathbf{u}(k+2) \\
\quad \oplus \cdots \oplus \mathbf{C} \mathbf{B}_{k+N-1} \otimes \mathbf{u}(k+N)
\end{array} \right. \quad (27)$$

この上記の式を現代制御理論の状態空間法と似た、簡潔な線形の式で表すと、

$$\mathbf{Y}(k+1) = \mathbf{\Gamma}_k \otimes \mathbf{x}(k) \oplus \mathbf{\Delta}_k \otimes \mathbf{U}(k+1) \quad (28)$$

となる。このときの行列  $\mathbf{Y}(k+1)$ ,  $\mathbf{U}(k+1)$ ,  $\mathbf{\Gamma}_k$ ,  $\mathbf{\Delta}_k$  は以下である。

$$\mathbf{Y}(k+1) = \begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N) \end{bmatrix} \quad (29)$$

$$\mathbf{U}(k+1) = \begin{bmatrix} u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N) \end{bmatrix} \quad (30)$$

$$\mathbf{\Gamma}_k = \begin{bmatrix} \mathbf{C}_{k+1} \otimes \mathbf{A}_k \\ \mathbf{C}_{k+2} \otimes \mathbf{A}_{k+1} \otimes \mathbf{A}_k \\ \vdots \\ \mathbf{C}_{k+N} \otimes \mathbf{A}_{k+N-1} \otimes \cdots \otimes \mathbf{A}_k \end{bmatrix} \quad (31)$$

$$\mathbf{\Delta}_k = \begin{bmatrix} \mathbf{C}_{k+1} \otimes \mathbf{B}_k & \cdots & \varepsilon \\ \mathbf{C}_{k+2} \otimes \mathbf{A}_{k+1} \otimes \mathbf{B}_k & & \vdots \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{k+N} \otimes \mathbf{A}_{k+N-1} \otimes \cdots \otimes \mathbf{A}_{k+1} \otimes \mathbf{B}_k & \cdots & \mathbf{C}_{k+N} \otimes \mathbf{B}_{k+N-1} \end{bmatrix} \quad (32)$$

モデル予測制御では望ましい出力を定め、これらの予測式を用いて最適な入力を求める。  
望ましい出力、つまり納期を、

$$\mathbf{r}(k+1) = \begin{bmatrix} r_1(k+i) \\ r_2(k+i) \\ \vdots \\ r_q(k+i) \end{bmatrix}, \quad \mathbf{R}(k+N) = \begin{bmatrix} \mathbf{r}(k+1) \\ \mathbf{r}(k+2) \\ \vdots \\ \mathbf{r}(k+N) \end{bmatrix} \quad (33)$$

と定める．納期までに外部へ出力していないといけないので，以下のように制約条件を表すことができる．

$$\mathbf{Y}(k+1) \leq \mathbf{R}(k+1) \quad (34)$$

$$\mathbf{\Gamma}_k \otimes \mathbf{x}(k) \oplus \mathbf{\Delta}_k \otimes \mathbf{U}(k+1) \leq \mathbf{R}(k+1) \quad (35)$$

この制約条件の下で，最適な入力を求める．

## 第 4 章 提案手法

本章では, 第 3 章で述べた max-plus 線形システムの未来の予測式を用いて最適化を行う. シンプレックス法を用いて最適化問題を解く. 最初に定式化しその後, 作業時間と部品到着時間とを分けて解く 2 段階法と, 同時に解く 1 段階法へと適用させていく.

### 4. 1 線形計画法への適用

システムの構造を表現するため, 以下のような行列を定義する.

$$\mathbf{P}_k := \text{diag}(\mathbf{d}(k)) \quad (36)$$

$$[\mathbf{F}]_{ij} := \begin{cases} e: \text{作業}i\text{が先行作業}j\text{を持つとき} \\ \varepsilon: \text{作業}i\text{が先行作業}j\text{を持たないとき} \end{cases} \quad (37)$$

$$[\mathbf{B}_0]_{ij} := \begin{cases} e: \text{作業}i\text{が外部からの入力}j\text{を持つとき} \\ \varepsilon: \text{作業}i\text{が外部からの入力}j\text{を持たないとき} \end{cases} \quad (38)$$

$$[\mathbf{C}]_i := \begin{cases} e: \text{作業}i\text{が外部への出力を持つとき} \\ \varepsilon: \text{作業}i\text{が外部への出力を持たないとき} \end{cases} \quad (39)$$

$\mathbf{P}$  を作業行列,  $\mathbf{F}$  を先行作業行列,  $\mathbf{B}$  を外部入力行列,  $\mathbf{C}$  を外部出力行列とそれぞれ呼ぶ. この時, max-plus 線形システムを表現する式の行列  $\mathbf{A}_k$  と  $\mathbf{B}_{k+1}$  は,

$$\mathbf{A}_k = (\mathbf{P}_k \otimes \mathbf{F})^* \otimes \mathbf{P}_k \quad (40)$$

$$\mathbf{B}_{k+1} = \mathbf{A}_k \otimes \mathbf{B}^T \quad (41)$$

と表すことができる. このとき  $(\mathbf{P}\mathbf{F})^*$  は,

$$(\mathbf{P}_k \otimes \mathbf{F})^* = e \oplus (\mathbf{P}_k \otimes \mathbf{F}) \oplus (\mathbf{P}_k \otimes \mathbf{F})^2 \oplus \dots \oplus (\mathbf{P}_k \otimes \mathbf{F})^l \quad (42)$$

である. ここで,  $l$  は先行作業の制約条件に依存する定数であり,  $(\mathbf{P}_k \otimes \mathbf{K})^l = \varepsilon$  である.

式(35)で与えた式のままでは, max-plus 代数の max 演算があり線形計画法へそのまま適用することができない. そこで各式の max 演算の部分を分解し, 線形計画問題の一般式に当てはめる.



## 4. 2 2 段階法

従来法である 2 段階法は、まず作業  $d_i$  に関する最適化問題を解いた後、外部からの部品の到着時刻に関する最適化問題を解く。2 段階法の第 1 ステップでは、部品の到着時刻に関しては考慮せず、作業時間についてのみの最適化を行う。この目的関数を作業時間のみを考慮した出力予測式が、納期と遅れの和より小さいという制約条件を持ち、式に表すと以下である。

$$\mathbf{C} \otimes (\mathbf{PF})^* \mathbf{P} \leq \mathbf{r}(k+1) + \boldsymbol{\mu}(k+1) \quad (43)$$

また作業時間には調整可能範囲が存在し以下の式で表せる。

$$0 \leq d_{min} < d_i < d_{max} \quad (44)$$

この制約条件の下で、遅れをできるだけ出さずに作業時間を長くするような目的関数を用いる。その式は以下の式で表せる。

$$J_1 = \alpha \cdot \max(\mu, 0) - \beta \cdot \sum_i d_i \rightarrow \min \quad (45)$$

ここで  $\alpha$  は遅れに対するペナルティを、 $\beta$  は作業時間を調整する際のコストを示す。

この線形計画問題を解いた後、第 2 ステップである外部からの部品の到着時刻に関する最適化問題を解く。納期と遅れの和から作業時間のみを考慮した出力予測時刻を差し引いた時刻より外部入力はや早いという制約条件を持つ。この制約条件を式に表したものが以下にある。

$$\mathbf{u}(k+1) \leq \mathbf{r}(k+1) + \boldsymbol{\mu}(k+1) - \{\mathbf{C}(\mathbf{PF})^* \mathbf{P} \otimes \mathbf{x}(k) \oplus \mathbf{C}(\mathbf{PF})^* \mathbf{P} \otimes \mathbf{B}_0\} \quad (46)$$

この制約条件のもとで、できるだけ部品の到着時刻が遅くなるような目的関数を用いる。その式は以下である。

$$J_2 = \alpha \cdot \max(\mu, 0) - \gamma \cdot \sum_j u_j \rightarrow \min \quad (47)$$

このとき  $\alpha$  は遅れに対するペナルティを、 $\gamma$  は部品の到着時刻を調整するコストを示す。

また外部からの部品のストックには在庫費がかかることが予測されるため、 $\beta \leq \gamma$ とする。これらの最適化問題は線形計画法を用いて解く。

### 4. 3 1 段階法

1 段階法は、作業  $d_i$  と外部からの入力  $u_i$  についての最適化問題を同時に解く。従来は作業時間と部品の到着時刻に関する最適化を 2 段階に分けて解くところを、1 段階で同時に解くことにより、2 段階法で生まれる可能性のあるロスを解消し、より優れた最適解を求めるのが目的である。

2 段階法と同様、 $(k+1)$  番目の出力の予測式  $\mathbf{y}(k+1)$  が、納期  $\mathbf{r}$  と遅れ  $\mu$  の和より早くなるという制約条件を持ち、式で表すと以下である。

$$\mathbf{y}(k+1) \leq \mathbf{r}(k+1) + \mu(k+1) \quad (48)$$

また作業時間の調整可能範囲は以下のように表せる。

$$0 \leq d_{min} < d_i < d_{max} \quad (49)$$

この制約条件の下、遅れをできるだけ出さないようにしつつ、作業にかかる時間の和と外部から部品の入力時刻の和を、できるだけ大きくするような目的関数を使用する。式に表すと以下になる。

$$J = \alpha * \max(\mu, 0) - \beta * \sum_i d_i - \gamma * \sum_i u_i \rightarrow \min \quad (50)$$

このとき  $\alpha$  は遅れ  $\mu$  に対するペナルティを意味し、 $\beta$  と  $\gamma$  はそれぞれ作業時間と部品の到着時刻の調整する際のコストを表している。目的関数値  $J$  は小さいほど、精度がよい。

# 第 5 章 数値実験

本章では前章で一般式に表した 1 段階法と 2 段階法とで数値実験を行う。数値実験の対象は、1 入力 1 出力  $n$  工程システムの  $(k+1)$  と、先行作業がランダムに決定される  $n$  工程のランダムグラフの  $(k+1)$  を予測し最適な生産スケジュールを構築する。

## 5. 1 様々なシステムでの数値実験

### 5. 1. 1 $n$ 工程の直列システム

$n$  工程の直列システムとは、図 2 に示したように 1 入力 1 出力で  $n$  個の工程を直列に並べたものである。数値実験では工程数を 1 つずつ増やし、その変化に伴う変動を考察する。

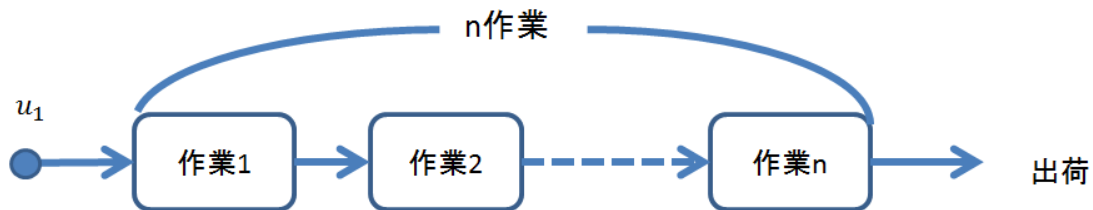


図 2 1 入力 1 出力  $n$  工程システム

このシステムの作業行列  $P$ ，先行作業行列  $F$ ，外部入力行列  $B_0$ ，外部出力行列  $C$  の各行列の成分を求めると次のように決定できる。

$$P = \text{diag}(d_1, d_2, \dots, d_n), F = \begin{bmatrix} \varepsilon & \varepsilon & \dots & \dots & \varepsilon \\ e & \varepsilon & & & \varepsilon \\ \varepsilon & e & \varepsilon & \dots & \varepsilon \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \varepsilon & \dots & \varepsilon & e & \varepsilon \end{bmatrix}, \quad (51)$$

$$B_0 = [e \quad \varepsilon \quad \dots \quad \varepsilon], C = [\varepsilon \quad \dots \quad \varepsilon \quad e] \quad (52)$$

納期  $r(k+1)$  は  $5 \cdot 5 \times (\text{工程数})$  と定めた。

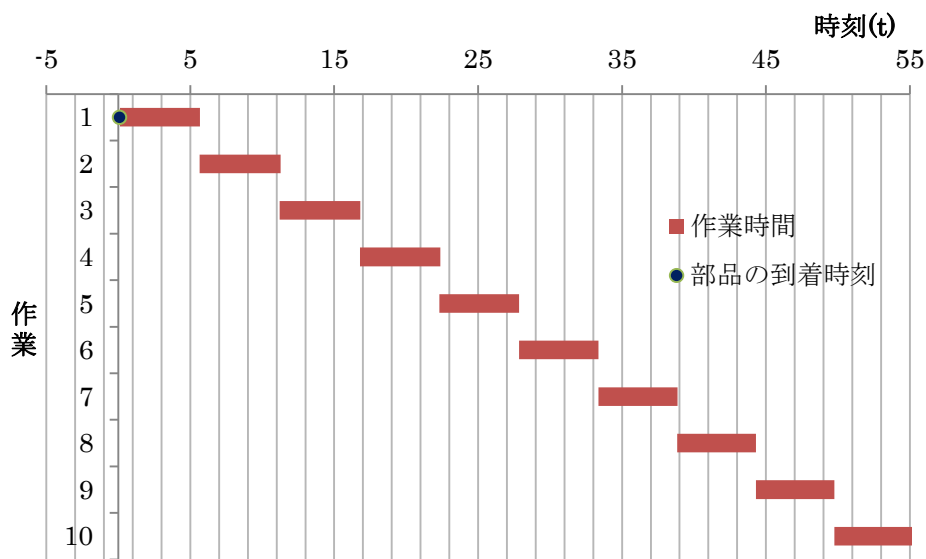


図 3 2段階法で求まる直列システムの生産スケジュール

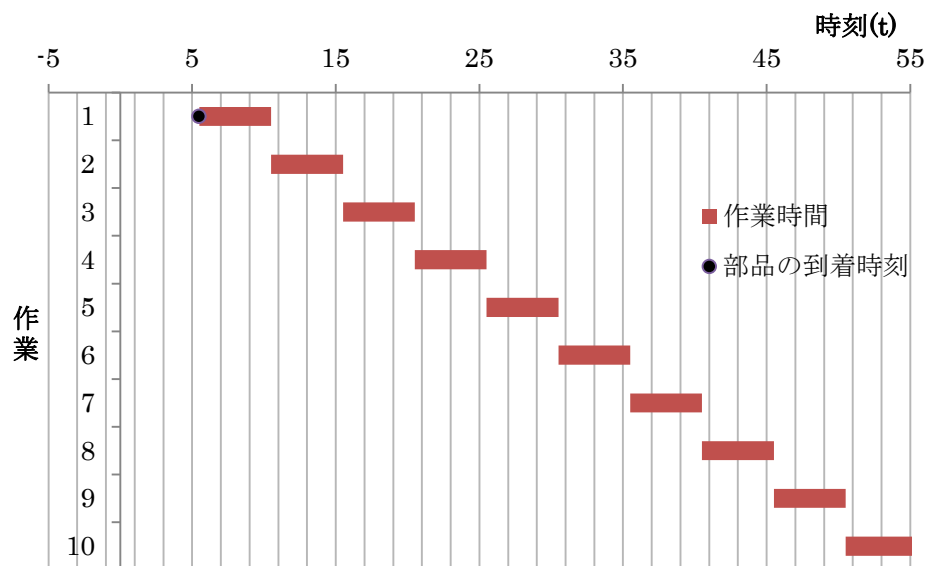


図 4 1段階法で求まる直列システムの生産スケジュール

図 3, 4 は 10 工程の直列システムの 2 段階法, 1 段階法でそれぞれ求まる生産スケジュール表である. 2 段階法では, 作業時間をできるだけ長くすることを採用しているのに対して, 1 段階法では作業時間を調整可能範囲いっぱいまで短くし, 部品の到着時刻を遅くすることを採用している. そのため, 1 段階法で求まる生産システム全体での作業時間は短くなっている.

## 5. 1. 2 ランダムでグラフを定めた場合

先行作業と外部からの入力ランダムに決定し、後続作業を持たない場合、その作業は外部へ出力を持つシステムについてとする。(k+1)番目のシステムの状態を予測し最適な各作業時間と、外部からの入力時刻を求める。

工程数 10, 外部からの部品の入力数 6 であり表 1 で示した先行作業と、外部からの部品の入力を持つ。

表 1 サンプルシステムの構造

作業	先行作業	外部からの入力	完成品の出力
1	-----	部品 1	
2	1	部品 6	
3	2	部品 2	
4	1	部品 1	
5	2, 3	-----	
6	1, 2, 4	部品 4	
7	1, 6	部品 3, 部品 5	
8	2, 6	部品 5	○
9	3, 5, 7	-----	
10	1, 5, 6, 7, 9	部品 6	○

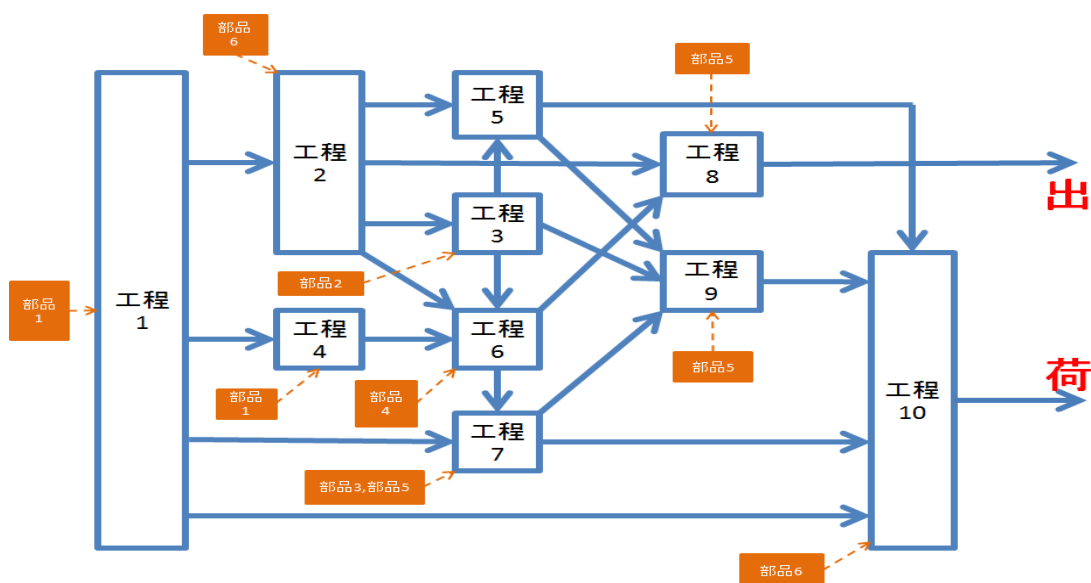


図 5 10 工程 6 入力のサンプルシステム

図 3 は表 1 に示したサンプルシステムを図に表したものである. このシステムの構造を表す行列  $F$ ,  $B_0$ ,  $C$  は以下のように定まる.

$$F = \begin{bmatrix} \varepsilon & & & & \dots & & & & & \varepsilon \\ e & \varepsilon & & & & & & & & \\ \varepsilon & e & \varepsilon & & & & & & & \\ e & \varepsilon & \varepsilon & e & & & & & & \\ \varepsilon & e & e & \varepsilon & \varepsilon & & & & & \vdots \\ e & e & \varepsilon & e & \varepsilon & \varepsilon & & & & \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & & & \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & & \\ \varepsilon & \varepsilon & e & \varepsilon & e & \varepsilon & e & \varepsilon & \varepsilon & \\ e & \varepsilon & \varepsilon & \varepsilon & e & e & e & \varepsilon & e & \varepsilon \end{bmatrix} \quad (53)$$

$$B_0 = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \quad (54)$$

$$C = [\varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad \varepsilon \quad e \quad \varepsilon \quad e] \quad (55)$$

このシステムの  $k+1$  番目を予測し, 最適化を行う. 作業時間の調整可能範囲は  $5 \leq d_i \leq 20$  とし, 納期  $r = 4 \times (\text{工程数})$  と定めた.

図 6, 7 は 2 段階法と 1 段階法のそれぞれで求めた作業のスケジュールである. 2 段階法では, 部品を早く到着させ作業時間を長くとることを採用している. 一方で 1 段階法では, 部品の到着時刻をできるだけ遅くし, 作業時間をタイトにしていることが分かる.

この生産システムでは部品 6 を工程 2 と工程 10 へ投入する. 2 段階法で求まる生産スケジュールでは, 工程 10 に部品 6 が到着してから作業開始までの待ち時間が約 25 である. 一方で, 1 段階法で求まる生産スケジュールでの部品の持ち時間は約 15 であり, 1 段階法では部品をストックしておくことで生まれる待ちコストの解消を実現している. また 2 段階法での, 外部へ完成品の出力を持つ作業 8 では, 納期より早く完成品ができ, 納期までの間の時間をロスしている. 一方, 1 段階法では, 外部への出力を持つ作業 8 と作業 10 は納期いっぱいまで作業を行っているため, 2 段階法で生まれたロスを解消している.

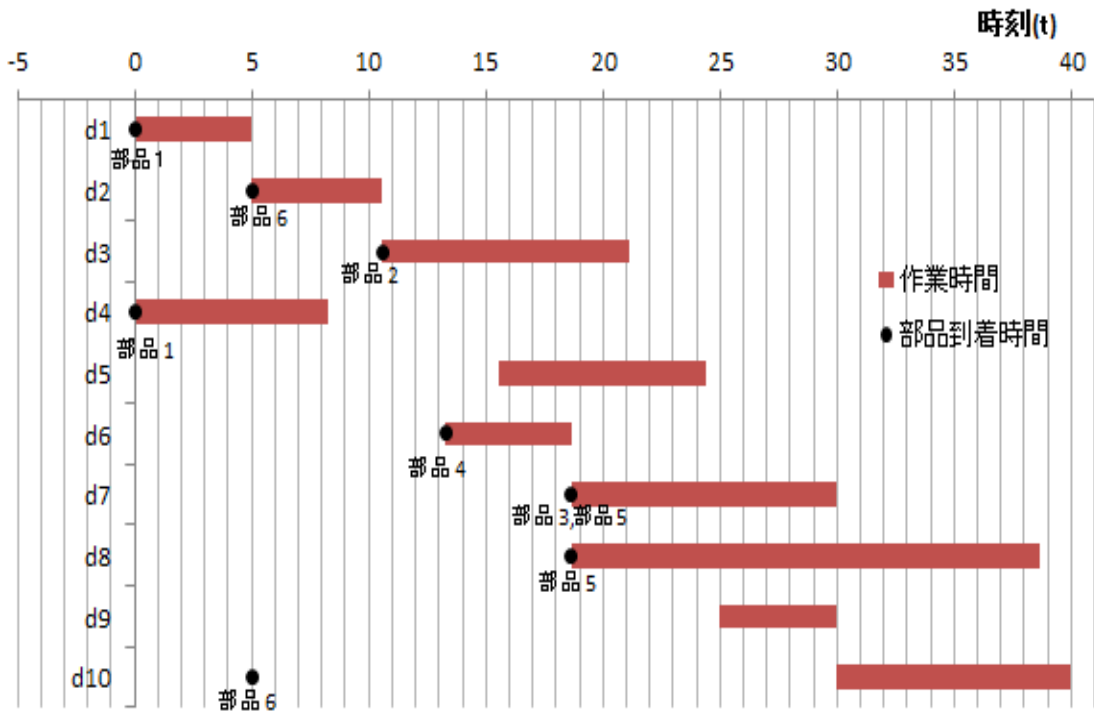


図 6 2段階法で求まる作業スケジュール

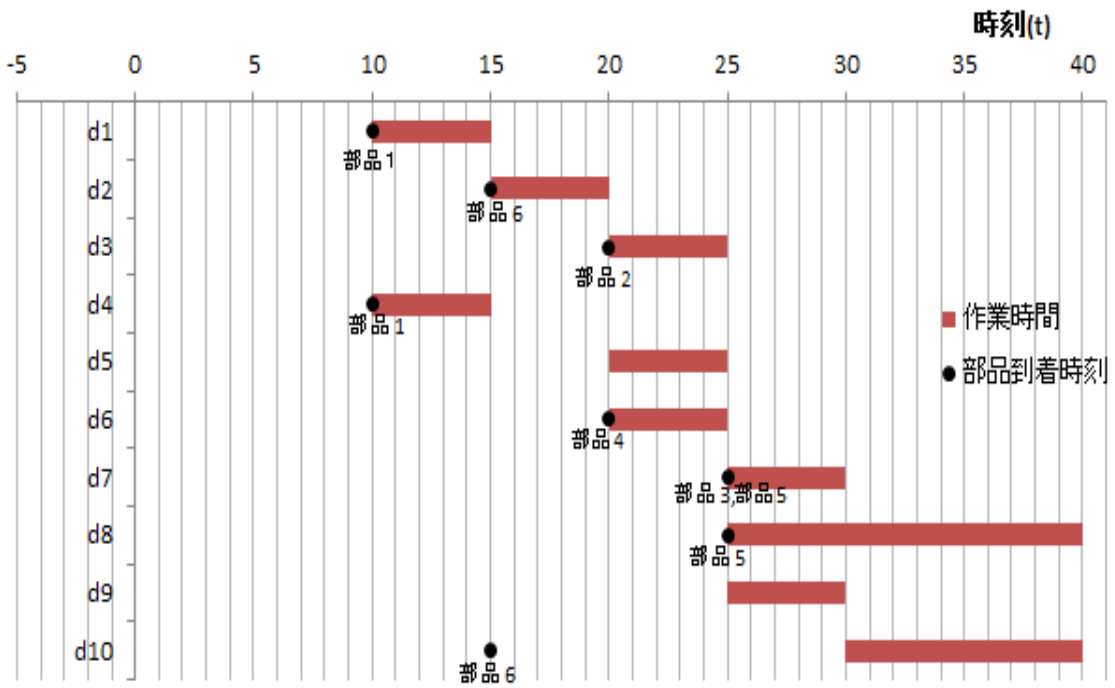


図 7 2段階法で求まる作業スケジュール

## 5. 2 従来法との性能の比較

本節では、数値実験での提案手法の1段階法と従来法の2段階法との目的関数値と計算時間、作業時間と部品の到着時刻の比較を行う。

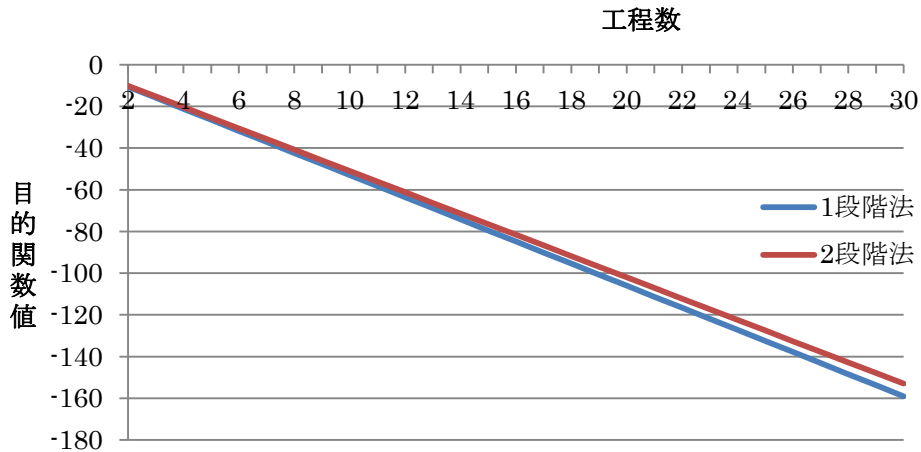


図 8 工程数と目的関数値の変化

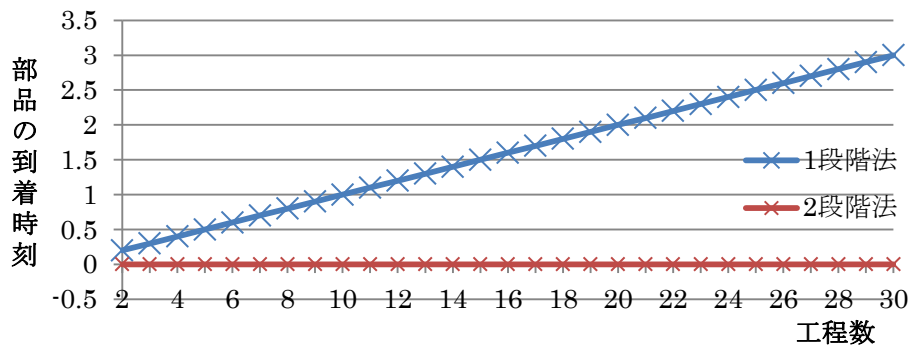


図 9 工程数と部品到着時刻の変化

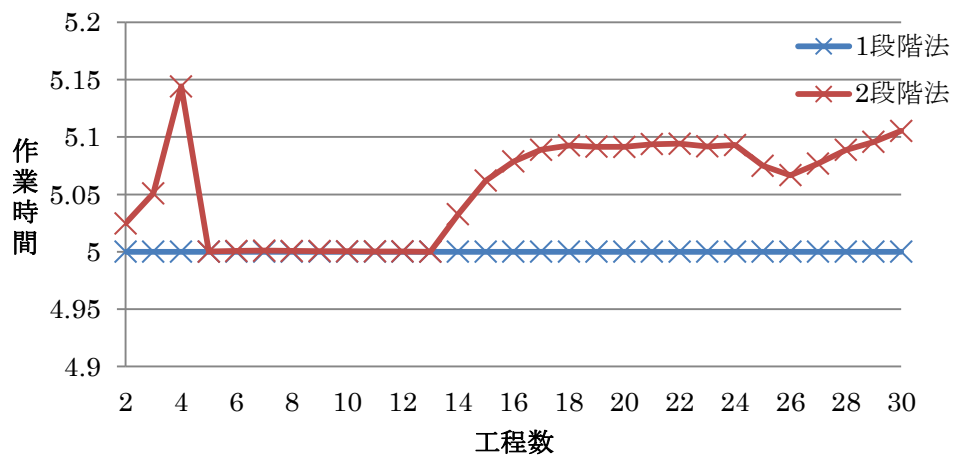


図 10 工程数と工程 2 の作業時間の変化



図8は1入力1出力の直列システムでの工程数の変化による目的関数値 $J$ の変化を表したものである。工程数が2のとき、目的関数値 $J$ の値は同じであるが、工程数が増えていくにつれて、2段階法より1段階法の方が優れた目的関数値が得られている。また、図9, 10はそれぞれ工程数の変化に伴う $d_2$ と $u_1$ の変化をそれぞれ示したグラフである。2段階法は作業時間を延ばすことを採用しているのに対して、1段階法は部品の到着時刻を遅くしていることを採用している。

図11は5.1.2で用いたサンプルシステムで、作業をランダムに選んで工程数を減らした時の目的関数値の変化である。工程4, 7, 6, 5, 3, 2, 8, 9の順番に工程を減らした。直列システムのように変化が直線ではないが、2段階法よりも1段階法で精度のよい目的関数値が得られている。

図12, 13はそれぞれ工程数の変化と工程2での作業時間の変化と部品1の到着時刻の変化を表したものである。2段階法では、作業時間をできるだけ大きくとり、システムに遅れが出そうなときに部品の到着時刻を早めることを採用している。一方で1段階法では作業時間を変化させないで、システムの構造にあわせて部品の到着時刻を変化させていることがわかる。

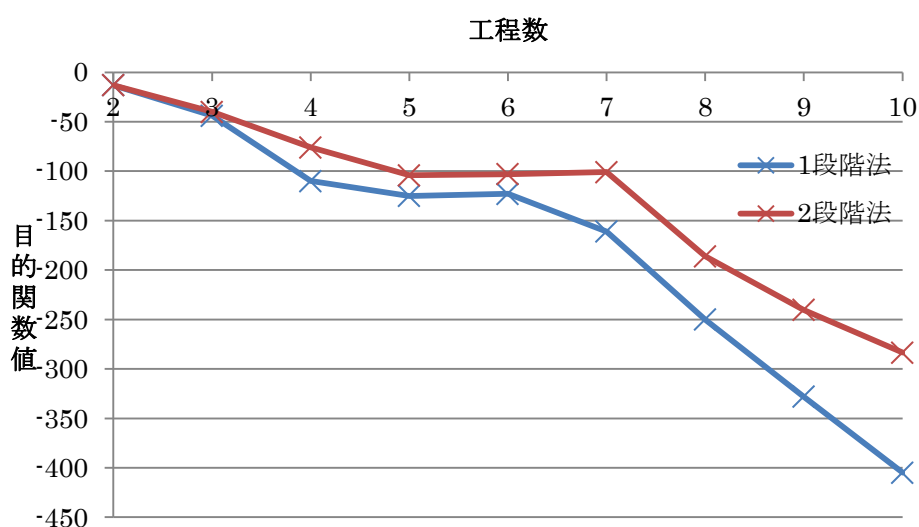


図11 サンプルシステムでの工程数と目的関数値の変化

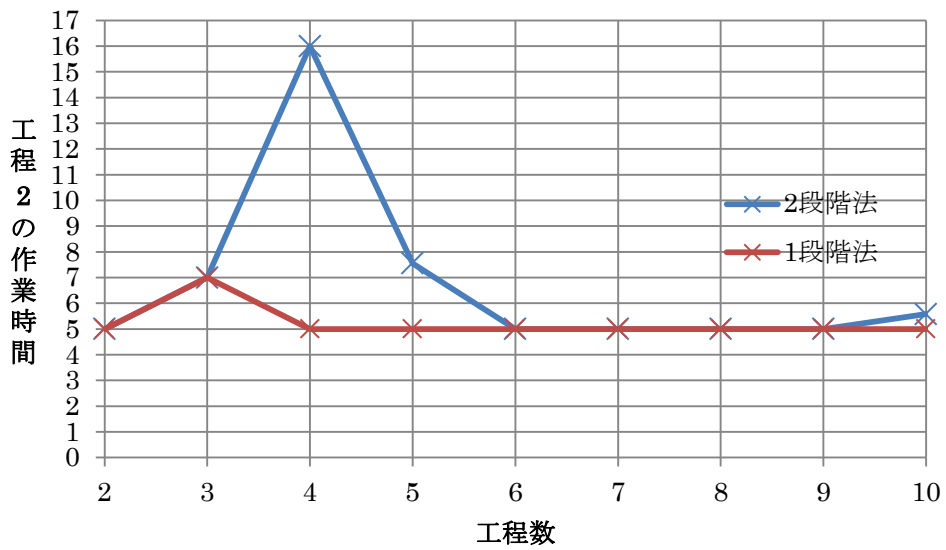


図 12 サンプルシステムでの工程数と工程 2 の作業時間の変化

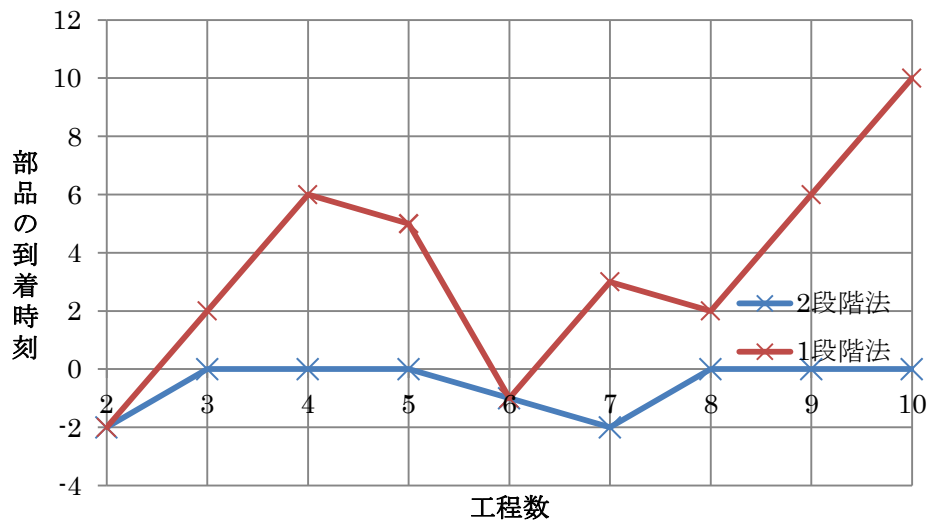


図 13 サンプルシステムでの工程数と部品 1 の到着時刻の変化

### 5. 3 評価

数値実験の結果、直列システムとランダムでシステムを作った場合のどちらも、2段階法より1段階法のほうが目的関数値の精度がよいことがわかった。また工程数が変化しても、1段階法の方が優れた目的関数値が得られた。

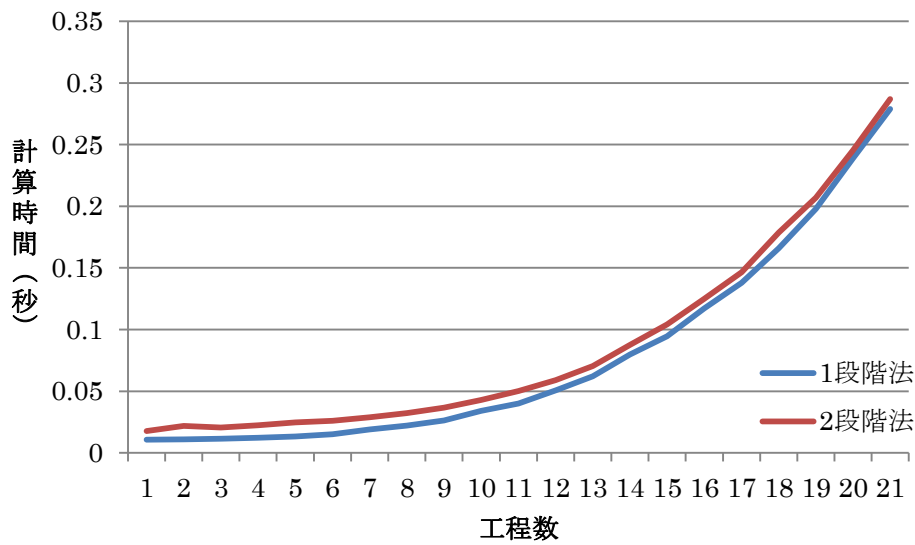


図 14 直列システムの計算時間

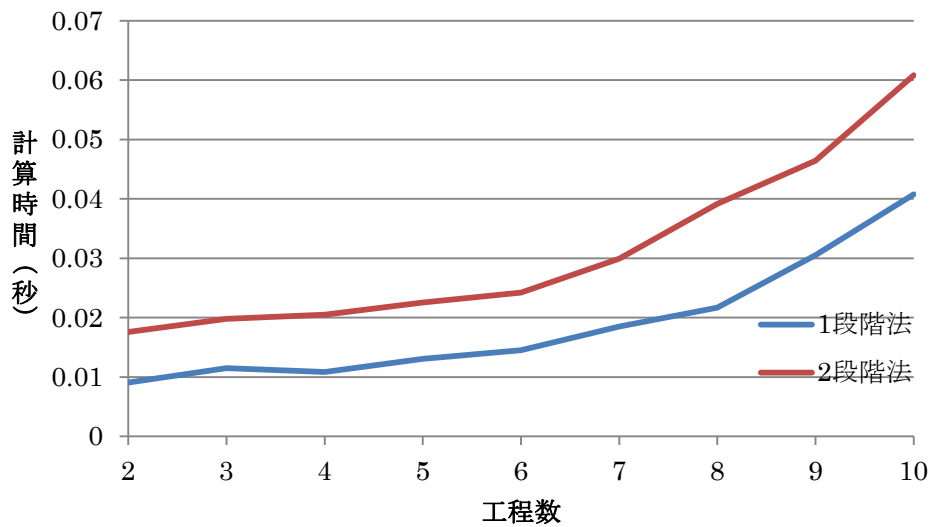


図 15 サンプルシステムの計算時間

図 14, 15 はそれぞれ直列システムとサンプルシステムの工程数を変えた時の最適化に要した計算時間の変化である。それぞれ最適化を 100 回行い、計算時間の平均をとった。1 段階法, 2 段階法ともに指数関数に似た計算時間の変化をしている。

直列システムの場合もサンプルシステムの場合も, 1 段階法でのシンプレックス法を 1 回で解くのに要する計算時間の方が, 2 段階法での部分最適化をシンプレックス法を用いて 2 解くのに要する時間より速いという結果が得られた。

## 第 6 章 結論

本研究では、企業における製造工程などの納期のあるシステムを主な研究対象とし、作業時間や部品の到着時刻を調整できる状況で、できるだけコストのかからないようにシステムを調整するしくみの構築を目的とした。そのために、**max-plus** 線形システムを用いてシステムの未来の状態予測し、その作業時間と部品の到着時刻についての最適化を行った。

従来は最初に作業時間についての最適化を行った後に、部品の到着時間の最適化を行う、2段階分けた最適化を行ってきた。しかし部分最適化を2回行うことで最適解にロスが生まれる可能性が想定され、そのロスを解消するために、作業時間と部品の到着時間の最適化を同時に行う1段階法を提案した。

本論文での提案手法の導入の効果は、次のようにまとめられる。

第一に、**max-plus** 線形システムを用いることで、多入力多出力システムでのシステムの内部の状態を簡単な線形表記で表すことができ、生産システムのスケジュールを組むことが可能になった。

第二に、本論文で提案した1段階法で得られた目的関数値は、数値実験において、どの場合においても2段階法と最適値が同じか2段階法より優れた値が求められた。

第三に、従来法では、作業時間を長くとる方法を採用していたため、部品の到着時刻が早まり、同じ部品を時間差で違う作業で使う場合に、部品をストックしておく時間や、完成品が2以上ある場合には、納期より早く完成品ができてしまうというケースがあり、時間にロスが生まれていた。本論文で提案した1段階法では、作業時間を短くし部品の到着時刻を遅くする方法を採用しており、部品や完成品をストックしておく時間を短くし、時間のロスを抑えることに成功している。

第四は、数値実験での計算時間の比較である。実験の結果、1段階法が2段階法と比べ最適化を行う際の計算時間が短いという結果が得られた。1段階法で1回分のシンプレックス法を解く計算時間より、2段階法で2回分のシンプレックス法を解く計算時間のほうが大きいことがわかった。

第一から第四をまとめると、**max-plus** 線形システムにおいて、作業時間と外部からの部品の到着時刻に関する最適化を同時に行う1段階法は有益な方法と言える。

## 参考文献

- [1] 森村英典, 刀根薫, 伊理正夫, OR 辞典, pp333-337, (2002)
- [2] B.A.Carre, An algebra for network routing problems, IMA Journal of Applied Mathematics.Volume 7,pp.273-294, (1971).
- [3] R.A.Cuninghame-Green, Minimax algebra, Lecture Note in Economics and Mathematical Systems,Springer-Verlag,Vol.166, (1979).
- [4] G.Cohen, D.Dubois, J.P.Quadrat, A linear-system-theoretic view of discrete-event processes and its use for performace evaluation in manufacturing, IEEE Trans, on Automatic Control, Vol. AC-30,No.3, pp.210-220, (1985).
- [5] 笠原統徳, クリティカルチェーン法を適用した max-plus 線形システム. (2010) .
- [6] 梶原宏之, 中尾洋一, Max-plus 代数に基づく製造ラインのスケジューリング問題の解法について, 日本船舶海洋工学会講演会論文集 第4号, (2007) .
- [7] 増田士朗, 五島洋行, Max-plus 線形システムによる離散事象システムモデリングとモデル予測制御. (2003) .
- [8] B. De. Schutter,T.J. J. van den Boom. Model predictive control for max-plus-linear systems, Automatica,Vol.37,No.7,pp.1049-1056, (2001) .
- [9] B.De.Schutter,T.J.J.van den Boom. MPC for perturbed max-plus-linear systems, Proc, of the European Control Conference 2001, pp.3783-3788, (2001).
- [10] Hiroyuki GOTO. Model predictive control-based scheduler for repetitive discrete event. (2012).
- [11] 雨宮孝, 竹安数博, 増田士朗, 新しい経営・経済数学, 中央経済社, (2004) .
- [12] B.De.Schutter, Max-algebraic system theory for discrete event systems, PhD thesis, Faculty of Applied Sciences, K.U.Leuven, (1996) .

## 謝辞

経営数理工学研究室に配属されてから、本論文を執筆するまでの期間、五島洋行 教授には、お忙しい中、貴重なお時間を割いていただき、丁寧なご指導をしていただきました。本研究を進行し、論文を執筆することができたのは、五島洋行 教授のおかげです。心より感謝申し上げます。

日々の研究生活で、多くの助言や励まし、協力をしていただいた経営数理工学研究室の仲間、先輩方に感謝します。

最後に、法政大学経営システム工学科で研究する機会を与え、暖かく見守ってくださった両親に深く感謝申し上げます。この場を持って、皆様への謝辞とさせていただきます。本当にありがとうございました。