

Max-min-plus scalingシステムでのモデル予測制御に基づくスケジューリング方法：滑走路のスケジューリングを例にして

豊田, 一裕 / TOYODA, Kazuhiro

(出版者 / Publisher)

法政大学大学院理工学・工学研究科

(雑誌名 / Journal or Publication Title)

法政大学大学院紀要. 理工学・工学研究科編 / 法政大学大学院紀要. 理工学・工学研究科編

(巻 / Volume)

57

(開始ページ / Start Page)

1

(終了ページ / End Page)

4

(発行年 / Year)

2016-03-24

(URL)

<https://doi.org/10.15002/00013375>

Max-min-plus scaling システムでのモデル予測制御に 基づくスケジューリング方法 ～滑走路のスケジューリングを例にして～

MODEL PREDICTIVE CONTROL-BASED SCHEDULER FOR MAX-MIN-PLUS SCALING SYSTEM
~A CASE OF A RUNWAY SCHEDULE~

豊田一裕

Kazuhiro TOYODA

指導教員 五島洋行

法政大学大学院理工学研究科システム工学専攻修士課程

In this research, we perform a scheduling based on model predictive control for max-min-plus scaling systems. There is a runway schedule as a typical example. We first determine the orders of takeoffs and landings of aircrafts using a combination of complete search and genetic algorithm. We then adjust the entry and occupation times for a runway with a method based on model predictive control for max-plus linear systems. Through a numerical simulation, we found that the proposed method can provide a schedule that satisfies the conditions of the max-min-plus scaling systems.

Key Words: scheduling, MPC, Max-min-plus scaling system, runway

1. はじめに

近年注目されている Just In Time を実現するのに有効な手段の一つがモデル予測制御 (Model Predictive Control: MPC) である。モデル予測制御の考え方は、望ましい目標値に出力を近づけるために、どのようなシステムの入力やシステム内部のパラメータを与えればよいかを求める手法である。現時点から数ステップ先の未来のシステムの状態を利用し、先読みした情報から早めに入力やシステムパラメータの調整ができ、優れたシステムの入力や状態が得られる。

本研究では、Max-min-plus scaling システムでのモデル予測制御に基づいたスケジューリングを行う。Max-min-plus scaling システムでのスケジューリング例として、空港の滑走路のスケジューリングが挙げられる。これまでモデル予測制御の理論を用いた滑走路のスケジューリングの研究は、Xiao-Bing Hu ら [1] が行っていたが、離陸機の考慮は行われていない。そこで本研究では、着陸機と離陸機の両方を考慮するため、max-min-plus system の条件を満たした滑走路のスケジューリング方法の提案を行う。

滑走路のスケジューリングでは、離陸機と着陸機の滑走路の使用順序を決めるときに max, plus 演算に加えて min 演算が必要であるが、これらの演算子を必要とするシステムでのモデル予測制御は非線形で非凸の問題である [2]。しかし離発着機の順序が決まっていれば、min 演算

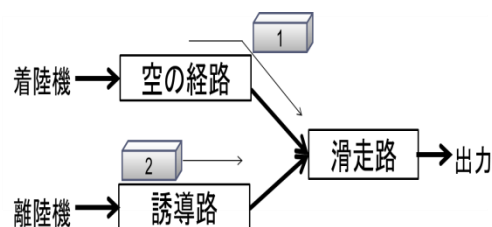


Fig. 1 max-min-plus scaling システムの例

が必要でない。Min 演算がなければ、max-plus 線形システムに基づいて滑走路のスケジューリングを表現できる。そこで本研究では、離着陸順を全数探索と遺伝的アルゴリズムで決め、滑走路の使用時間や滑走への侵入時刻は、max-plus 線形システムでのモデル予測制御に基づいたスケジューリング手法で調整する。

2. MMPS での MPC

(1) Max-min-plus scaling システム

Max-min-plus scaling システム (MMPS) とは、max-plus 代数 [3] の延長で、max 演算、min 演算、+ 演算を基本演算とする代数系を用いてモデル化されるシステムモデルのことである。主に離散事象システムで、事象の発生回数によるシステムの振る舞いをモデル化することに適しており、B. De Schutter [4] によってモデル予測制御 (MPC) への応用が行われている。ここで扱う max-plus 代数とは、max 演算、+ 演算を基本演算とする

代数系のことであり、以下のように定義する.

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

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

\oplus , \otimes の単位元はそれぞれ $-\infty$, 0 と定義され、それぞれ ε と e と表記する. これらの演算子で定義される代数系は、行列演算やベクトルの演算でも同様に計算できる.

MMPSの例をFig. 1を用いて説明する. 空の経路と誘導路から滑走路へ機体を侵入させるときは、滑走路への到着が早い方を優先させる. 例として着陸機1が先に滑走路へ行き、その後で離陸機2が滑走路へ行くというように、滑走路の使用順序が決まっている場合、以下の式で表せる.

$$\begin{aligned} x_1(1) &= \max\{x_1(0), u_1(1)\} + d_1(1) \\ x_2(1) &= \max\{x_2(0), u_2(1)\} + d_2(1) \end{aligned} \quad (3)$$

$$\begin{aligned} x_3(1) &= \max\{x_3(0), x_1(1)\} + d_3(1) \\ x_1(2) &= \max\{x_1(1), u_1(2)\} + d_1(2) \\ x_2(2) &= \max\{x_2(1), u_2(2)\} + d_2(2) \\ x_3(2) &= \max\{x_3(1), x_2(2)\} + d_3(2) \end{aligned} \quad (4)$$

ここで $x_i(k)$ の k は、事象の発生回数を表し、イベントカウンタと呼ばれる. この例の場合、着陸機1は $k=1$ 、着陸機2は $k=2$ である. また初期状態を $x_i(0) = \varepsilon$ と表す. 滑走路への侵入経路の使用時間を $d_1(k)$ 、離陸機の誘導路の使用時間を $d_2(k)$ 、滑走路の使用時間の時間を $d_3(k)$ 、滑走路への侵入経路への入力時刻は $u_i(k)$ 、誘導路への入力時刻は $u_i(k)$ である. また着陸機1の関係のない誘導路の使用時間は $d_2(1) = 0$ 、誘導路への入力時刻は $u_2(1) = \varepsilon$ である. 同様に、離陸機2の関係のない空の滑走路への侵入経路の使用時間と、誘導路への入力時刻はそれぞれ $d_2(1) = 0$ 、 $u_2(1) = \varepsilon$ である.

一般的にmax-min-plus scalingシステムでのモデル予測制御は、非線形問題で非凸の問題であることが知られているが、例のように離着陸順が決まっていれば、max-plus線形システムでのモデル予測制御として考えることができる. しかし、一般的に離着陸順は決まっていないため、離着陸順の全てのパターンを調べる全数探索と、短時間で近似解を得る遺伝的アルゴリズムを用いて離着陸順を決定する.

(2) 文字の定義

本研究で扱う文字、ベクトル、行列の定義をする.

n : 状態数

m : 外部からの入力数

N_p : 滑走路を使用する飛行機の総数

N : 予測ステップ数

$\mathbf{d}(k) \in \mathbb{R}_{\max}^{m \times n}$: ステップ k での状態の時間のベクトル.

$\mathbf{u}(k) \in \mathbb{R}_{\max}^m$: ステップ k での外部からの入力時刻のベクトル.

$\mathbf{x}(k) \in \mathbb{R}_{\max}^n$: ステップ k での各状態の終了時刻のベクトル.

$\mathbf{y}(k) \in \mathbb{R}_{\max}^m$: ステップ k での離発着時刻のベクトル.

$\mathbf{r}(k) \in \mathbb{R}^N$: ステップ k での離発着予定時刻のベクトル.

$\mathbf{P}_k \in \mathbb{R}_{\max}^{n \times n} := \text{diag}\{d_i(k)\}$

$[\mathbf{w}_k]_i \in \mathbb{R}_{\max}^n := \begin{cases} e: k\text{番目の機体が状態}i\text{を通る} \\ \varepsilon: k\text{番目の機体が状態}i\text{を通らない} \end{cases}$

$\mathbf{W}_k = \text{diag}(\mathbf{w}_k)$

$[\mathbf{F}_0]_{ij} \in \mathbb{R}_{\max}^{n \times n} := \begin{cases} e: \text{状態}i\text{が先行状態}j\text{を持つ} \\ \varepsilon: \text{状態}i\text{が先行状態}j\text{を持たない} \end{cases}$

$[\mathbf{B}_0]_{ij} \in \mathbb{R}_{\max}^{n \times m} := \begin{cases} e: \text{状態}i\text{が外部からの入力}j\text{を持つ} \\ \varepsilon: \text{状態}i\text{が外部からの入力}j\text{を持たない} \end{cases}$

$[\mathbf{C}_0]_i \in \mathbb{R}_{\max}^n := \begin{cases} e: \text{状態}i\text{が外部への出力を持つ} \\ \varepsilon: \text{状態}i\text{が外部への出力を持たない} \end{cases}$

ここで、 \mathbf{B}_0 と \mathbf{C}_0 はそれぞれ、入力行列、出力行列と呼ばれる.

(3) 予測式の導出

イベントカウンタ k のとき、通り道のみ情報が入った先行関係の行列は以下である.

$$\mathbf{F}'_k = \mathbf{W}_k \otimes \mathbf{F}_0 \otimes \mathbf{W}_k \quad (5)$$

このとき、

$$(\mathbf{F}'_k \otimes \mathbf{P}_k)^* = e \oplus (\mathbf{F}'_k \otimes \mathbf{P}_k) \oplus (\mathbf{F}'_k \otimes \mathbf{P}_k)^2 \oplus \dots \oplus (\mathbf{F}'_k \otimes \mathbf{P}_k)^l \quad (6)$$

$$\mathbf{A}'_k = \mathbf{P}_k \otimes (\mathbf{F}'_k \otimes \mathbf{P}_k)^* \quad (7)$$

$$\mathbf{B}'_{k+1} = \mathbf{P}_k \otimes (\mathbf{F}'_k \otimes \mathbf{P}_k)^* \otimes \mathbf{W}_k \otimes \mathbf{B}_0 \quad (8)$$

$$\mathbf{C}'_k = \mathbf{C}_0 \otimes \mathbf{W}_k \quad (9)$$

とすると、状態予測式、出力予測式は以下のように表せる.

$$\mathbf{x}(k+1) = \mathbf{A}'_k \otimes \mathbf{x}(k) \oplus \mathbf{B}'_{k+1} \otimes \mathbf{u}(k+1) \quad (10)$$

$$\mathbf{y}(k+1) = \mathbf{C}'_k \otimes \mathbf{x}(k+1) \quad (11)$$

そして、式(10)、(11)の右辺の k を $k+1$ にすることを繰り返してい、 N ステップ先の状態を予測すると以下の式で表せる.

$$\begin{aligned} \mathbf{x}(k+N) &= \mathbf{A}'_{k+N-1} \otimes \dots \otimes \mathbf{A}'_k \otimes \mathbf{x}(k) \\ &\oplus \mathbf{A}'_{k+N-1} \otimes \dots \otimes \mathbf{A}'_k \otimes \mathbf{B}'_k \otimes \mathbf{u}(k) \\ &\oplus \dots \oplus \mathbf{B}'_{k+N-1} \otimes \mathbf{u}(k+N) \end{aligned} \quad (12)$$

さらに、

$$\mathbf{\Gamma}'_k = \begin{bmatrix} \mathbf{C}'_{k+1} \otimes \mathbf{A}'_k \\ \mathbf{C}'_{k+2} \otimes \mathbf{A}'_{k+1} \otimes \mathbf{A}'_k \\ \vdots \\ \mathbf{C}'_{k+N} \otimes \mathbf{A}'_{k+N-1} \otimes \dots \otimes \mathbf{A}'_k \end{bmatrix}$$

$$\mathbf{\Delta}'_{k+1} = \begin{bmatrix} \mathbf{C}'_{k+1} \otimes \mathbf{B}'_{k+1} & \dots & \varepsilon \\ \mathbf{C}'_{k+2} \otimes \mathbf{A}'_{k+1} \otimes \mathbf{B}'_{k+1} & \dots & \vdots \\ \vdots & \ddots & \varepsilon \\ \mathbf{C}'_{k+N} \otimes \mathbf{A}'_{k+N-1} \otimes \dots \otimes \mathbf{A}'_{k+1} \otimes \mathbf{B}'_{k+1} & \dots & \mathbf{C}'_{k+N} \otimes \mathbf{B}'_{k+N-1} \end{bmatrix}$$

を用いると各ステップの状態と出力予測式は、

$$\mathbf{Y}(k+1) = \mathbf{\Gamma}'_k \otimes \mathbf{x}(k) \oplus \mathbf{\Delta}'_{k+1} \otimes \mathbf{U}(k+1) \quad (13)$$

である.

(4) 制御目標

各ステップでの出発時刻と到着時刻に対しての遅れを、

$$\mu(k+l) = \max\{y(k+l) - r(k+l), 0\} \quad (14)$$

とする。各ステップで出力予測時刻が発着時刻と遅れの和より小さくならなければいけない。そのため、ステップ $k+N$ までの制約条件は、式(13)を用いて以下のように与えられる。

$$Y(k+1) \leq R(k+1) + \bar{\mu} \quad (15)$$

ここで、

$$R(k+1) = [r^T(k+1) \quad r^T(k+2) \quad \dots \quad r^T(k+N)]^T \quad (16)$$

$\bar{\mu} = [\mu^T(k+1) \quad \mu^T(k+2) \quad \dots \quad \mu^T(k+N)]^T$ (17) である。また各状態の調整可能範囲として以下の制約条件を与える。

$$0 < d_{\min} \leq d_i(k+l) \leq d_{\max} \quad 1 \leq i \leq n \quad (18)$$

次に滑走路の使用時間を長く、遅れをできるだけ出さないことを考える。本研究では、より無駄の少ないスケジュールを組むため、誘導路と滑走路の使用時間 d_i と誘導路への入力時刻 u_k の最適化を同時に行う。遅れを出さず、誘導路の使用時間 d を小さく、外部からの入力時刻 u を大きくすることを考える。これらを定式化すると以下の式で表せる。

$$J_1 = \alpha \sum_{j=1}^N \mu(k+j) + \beta \sum_{j=1}^N \sum_{i=1,2}^n d_i(k+j) - \gamma \sum_{j=1}^N \sum_{i=1,2}^m u_i(k+j) \quad (19)$$

ここで、 α は遅れに対するペナルティを、 $\beta, \gamma (\geq 0)$ はそれぞれ状態の時間と外部入力調整パラメータである。線形計画問題を用いて、 $d(k+l), u(k+l), \mu(k+l)$ について同時に最適化する最適化問題は、以下のように表せる。

$$J_1 \rightarrow \min \quad \text{s.t.} \quad \text{Eqs.}(15),(18) \quad (20)$$

3. 順序決め

(1) 全数探索を用いた順序決め

全数探索は、離着陸順の全てのパターンを調べて、その中で最も式(19)の値が小さいときを最適解とする。離着陸順の全てを調べるため、組合せの数は予測ステップ数が $N=8$ のときは $8! = 40,320$ 通り、 $N=10$ のときは $10! = 3,628,800$ 通りである。

(2) 遺伝的アルゴリズムを用いた順序決め

全数探索を用いて厳密解を得るには、非常に長い計算時間が必要である。モデル予測制御は、未来の予測をしながら、1ステップ先の入力やシステムパラメータを決めるものであり、計算に時間がかかると間に合わない。そのため良い解を比較的短時間で求める必要がある。そこで本研究では、遺伝的アルゴリズム(GA)を用いて近似解を求める。GAには選択、突然変異、交叉の3種類の遺伝的操作が用いられる。その中でも交叉は、実行不能解を生成しないような適切な方法を用いる必要がある[5]。本

研究では、主に巡回セールスマン問題で用いられる順序交叉、周期交叉を用いて、離発着の順序を決定する。

a) 順序交叉(OX)

順序交叉は、一つの親から順列の一部を受け継ぎ、残りの部分をもう一つの親の順列の位置を受け継ぐ方法である。順序交叉のアルゴリズムは以下である。

- i. 親を2点で切断する。
- ii. 切断した中央部を子に受け継ぐ。
- iii. 中央部を受け継いだ親とは別の親の順列の後ろの切断点から開始し、要素の系列を作成する。最後の要素になったら先頭から繰り返す。ただし、すでに子に含まれている値は含まない。
- iv. 3で求めた系列順に、後ろの切断点から順に残りの要素を埋める。

b) 周期交叉(CX)

周期交叉は親の順列の相対位置を受け継ぐ方法である。周期交叉のアルゴリズムは以下である。

$p_1 = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8)$, $p_2 = (8\ 4\ 7\ 2\ 1\ 6\ 5\ 3)$ について考える。

- i. p_1 の最初の要素1から開始して、 p_2 の1番目の要素は8なので、 c_1 に8を受け継ぐ。 p_2 の8番目の要素は3なので、 c_1 に3を受け継ぐ。これらの手順を繰り返すと c_1 は以下のように得られる。

$$c_1 = (1\ X\ 3\ X\ 5\ X\ 7\ 8)$$

- ii. 最後に c_1 に入った値は5であるが、この位置に対応する p_2 の値は1である。これ以上は進めないため、残りの要素は p_2 からそのまま受け継ぐ。

$$c_1 = (1\ 4\ 3\ 2\ 5\ 6\ 7\ 8)$$

- iii. p_2 に対しても同様の手順を踏むと c_2 は以下のように得られる。

$$c_2 = (8\ 2\ 7\ 4\ 1\ 6\ 5\ 3)$$

4. 数値実験

(1) ケーススタディ

空港の誘導路と滑走路を Fig. 2 ようにモデル化する。Fig. 2 の1~5は離陸機が滑走路までの誘導路を、6は着陸機が滑走路に進入する経路を、Task7は滑走路を、8は着陸機がゲートまで通る誘導路を、9は離陸後の経路をそれぞれ示す。滑走路のスケジュールを求めるための入力と出力は以下である。

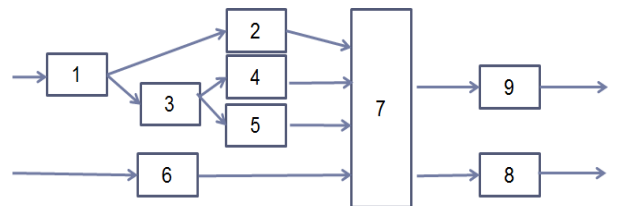


Fig. 2 ケーススタディのモデル図

入力 :

- 先行関係 : $F_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & e & e & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix}^T$
- 入力行列 : $B_0 = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}^T$
- 予測ステップ数: $N=8$

出力 :

- 誘導路の使用時間 : $d(k+l)$
- 各入力時刻 : $u(k+l)$

(2) 実験結果

Fig. 3 は提案手法による滑走路のスケジュールである。色付けの数字は Fig. 2 の状態の値と一致する。GA を用いて離発着順を決定した場合も、同じスケジュールを得た。飛行機 1 の誘導路への入力時刻が、飛行機 3 よりも早い。飛行機 3 の方が先に滑走路に到着し、先に滑走路を使用している。これらのことから、max-min-plus scaling システムの条件を満たしているスケジュールが組み立てることがわかる。Fig. 4 は、交叉方法による解の改善の過程を示したものである。どちらの手法も、交叉率 30%、突然変異率 30% で数値実験を行った結果である。OX を用いたほうが速く最適解が得られた。Fig. 5 は各手法の計算時間を示したものである。GA は世代数が 100、個体数が 20 の場合の計算時間である。全数探索の計算時間は指数分布に従うように増えている。一方で GA の計算時間は全数探索に比べ緩やかに増加している。ケーススタディ

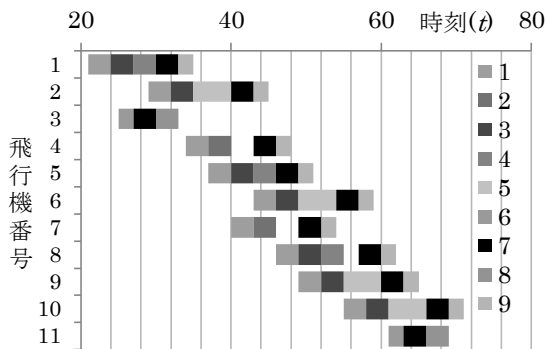


Fig. 3 ケーススタディのスケジュール

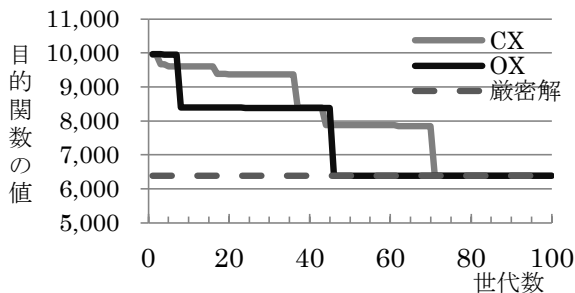


Fig. 4 GA による改善過程

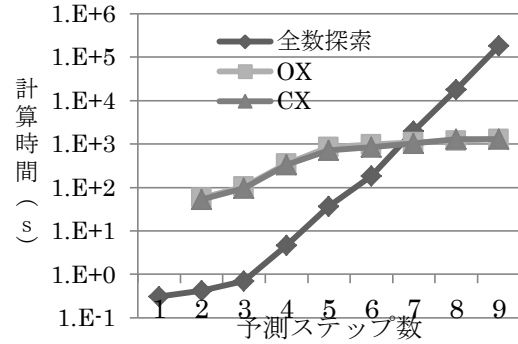


Fig. 5 各手法の計算時間

での数値実験より、予測ステップ数が 7 未満のときは全数探索を、7 以上のときは GA を用いて離発着順を決めた方がよいことがわかった。

5. おわりに

本研究では、Max-min-plus scaling システムでのモデル予測制御を行った。一般的に MMPS でのモデル予測制御は、非線形問題であるため、本研究では離着陸順を全数探索と GA を用いて決め、滑走路の使用時間や滑走への侵入時刻は、max-plus 線形システムでのモデル予測制御に基づいたスケジューリング手法で調整した。ケーススタディを用いた数値実験の結果、提案手法を用いて MMPS の条件を満たすスケジュールが得られた。また予測ステップ数が 7 未満のときは全数探索を、7 以上のときは GA を用いて離発着順の決定を行った方がよいことがわかった。

参考文献

- 1) Xiao-Bing Hu, and Wen-Hua Chen: "Receding horizon control for aircraft arrival sequencing and scheduling", Institute of Electrical and Electronics Engineers, vol.6, pp189-197, 2005
- 2) B. De Schutter, and T. van den Boom: "Model predictive control for max-min-plus scaling systems", Proceedings of the 2001 American Control Conference, pp.325-330, 2001.
- 3) H. Goto: "Model predictive control-based scheduler for repetitive discrete event systems with capacity constraints, An International Journal of Optimization and Control: Theories & Applications, vol.3, no.3, pp.73-83, 2013.
- 4) B. De Schutter, and T. van den Boom : "Model predictive control for max-min-plus scaling systems - Efficient implementation", Proceeding of the 6th International Workshop on Discrete Event Systems, pp.343-348, 2002
- 5) S. Matsui, S. Yamada: 'A genetic algorithm for optimizing hierarchical menus", Institute of Electrical and Electronics Engineers C, IEEE Congress on Evolutionary Computation 2008, pp.2851-2858, 2008