

# 先端生命維持システム運用のためのスケジューリングアルゴリズムの開発

Development of a Scheduling Algorithm for Advanced Life Support System Operation

○宮嶋宏行 (東京女学館大)、広崎朋史 (宇宙システム開発)、石川芳男 (日大)

Hiroyuki Miyajima\*, Tomofumi Hirotsaki\*\*, Yoshio Ishikawa\*\*\*

\* Tokyo Jogakkan College, 1105 Tsuruma, Machida-shi, Tokyo 194-0004, Japan

E-mail : miyajima@m.tjk.ac.jp

\*\* Space Systems Development Corporation, 2-10-1-202 Minamihanazono, Hanamigawa-ku, Chiba-shi, Chiba 262-0022, Japan

\*\*\* Nihon University, 7-24-1 Narashinodai, Funabashi-shi, Chiba 274-8501, Japan

## ABSTRACT

A habitation experiment using the Closed Ecology Experiment Facilities was started in 2005. In the future, the stays will be gradually extended. We have been developing the three layered control software for a Control Computer System of the Closed Ecology Experiment Facilities in order to back up the habitation experiments. In this paper, we will show the development of an operation scheduling system for one of the three layers, such as at the planning and scheduling level, and discuss the development of a scheduling algorithm that does not cause the complexity of the ALS scheduler to be exponentially increased.

**Key Word** : Three Layers Control, Lagrange Relaxation, Lagrange Decomposition and Configuration , Dynamic Scheduling

## 1. はじめに

2005 年度から閉鎖型生態系実験施設(Closed Ecology Experiment Facilities : CEEF)を利用した居住実験が始まり、今後、滞在期間は徐々に延ばされ実験が行われる<sup>1)</sup>。前報<sup>2)</sup>では居住実験を支援するための Fig.1 に示すような CEEF 制御用計算機システムの階層制御ソフトウェアについて述べた。本論文では、そのうちの 1 層のプランニング&スケジューリングレベルにおける運用スケジュール作成システム (以後これを ALS スケジューラと呼ぶ) の開発について示し、ALS スケジューラのための計算量が指数的に増加しないスケジューリング・アルゴリズムの開発について論じる。

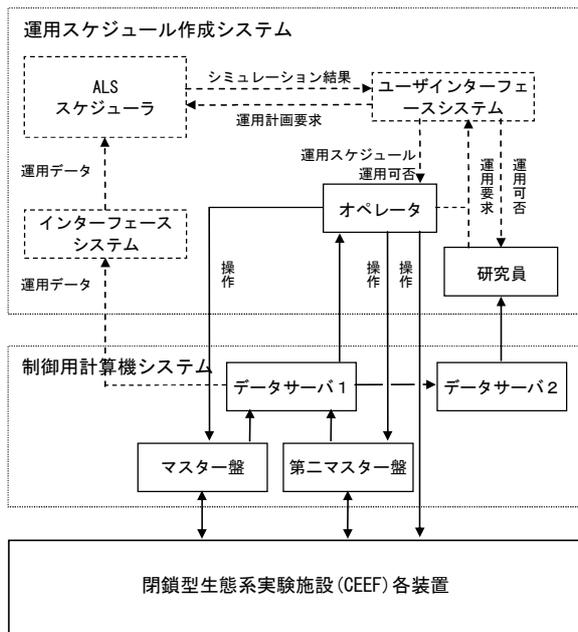


Fig.1 Expanded control computer system of the CEEF

ALS スケジューラは MS Visual C++ 2005 を利用して開発され、Fig.2 に示すようなソフトウェアとなる予定である。ALS スケジューラは、PSL(Planning and Scheduling Language)によってスケジューリングモデルを記述する機能、スケジューリング・アルゴリズムによりジョブを割り付ける機能、その結果をガントチャートと状態量変化グラフとして表示する機能を持つ。オペレータはこのスケジューラを用いて実験開始前に実験計画に基づいた運用スケジュールを作成し、その結果を確認する。また、実験開始後に実験計画の変更や機器の異常が起こった場合には運用データを基に運用スケジュールを再作成し、その結果を確認する。作成された運用スケジュールは、オペレータの操作により実行される。

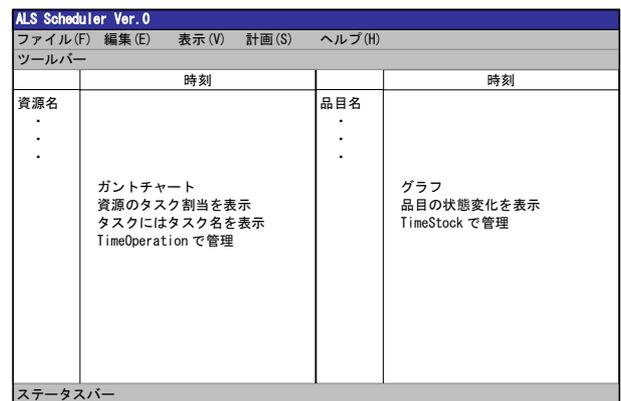


Fig.2 Screen of the ALS scheduler

さて、スケジューリング問題は組合せ問題であり、在庫に対する時間最適化問題を解くことである。これは NP 困難あるいは NP 完全と呼ばれる問題に属する。この問題では計算時間が問題の規模に応じて指数的に増加する(Bellman の次元の呪い)。従来のスケジューリングでは数値計画法 (総当り法、動的計画法、分枝限定法など)

やメタヒューリスティクス (GA,SA,TS,問題に固有のヒューリスティクス)、ディスパッチング・ルールが利用されていた。数理計画法は、次元が大きくなると解くことが難しくなり、メタヒューリスティクスも、ある種の問題に対しては適当な時間で準最適解を求めることができるが、問題ごとに定式化の工夫やパラメータの調整が必要である。ディスパッチング・ルールは、従来最も利用されていた方法で、計算に無駄がなく、どのような問題にも適用できる頑強なスケジューリングであるが、そのルールの獲得は経験によるところが大きい。

ALS スケジューラの開発では、大規模な問題に対応できる動的スケジューリングのアルゴリズムの実装を目指す。動的スケジューリングとは、スケジューリングに関するデータの一部がスケジューラ作成時に未知であったり、スケジューラ作成後に変更されたりする問題である。このような動的スケジューリングに最適化手法を用いることは、計算速度やコンピュータ処理能力の点から今までは現実的ではなかった。動的スケジューリングに最適化手法を用いるためには、スケジューリング問題の指数的な計算量の増加を克服し実用的な時間で解を探索する解法が必要となる。それによって、大規模なシステムの動的スケジューリングが可能になる。

そこで、本研究では計算量が指数的に増加しないスケジューリング・アルゴリズムの開発を目的とする。次章以降では、ラグランジュ分解・調整法による定式化、ラグランジュ乗数の決定法、動的スケジューリングへの適用の可能性の3点について論じる。

## 2. スケジューリング・アルゴリズム

スケジューリング問題は計画期間に対して指数的に計算量が増加していく。動的スケジューリングを考えた場合、その計算量の増加が問題となる。これを克服するスケジューリングの最も有力な方法の1つがラグランジュ分解・調整法である<sup>3)</sup>。本方法は分解法を利用することで、計算量の指数的な増加を抑えながら、実用的な時間で準最適解を求めることができる。この方法では、あるジョブの割り付けを他のジョブの割り付け状況を見捨て、それぞれのジョブにとって最も望ましいタイムスロット(スケジューリングの時間間隔)を選んで割り付ける。ジョブの競合が発生した場合には平準化を行なって競合を解消する。

次に、ラグランジュ分解・調整法を用いて ALS システムのスケジューリング問題を定式化する。

### 2.1 ラグランジュ分解・調整法による定式化

記号と添字を次のように定義する。

$i$ : 状態番号 ( $i=1,2,\dots,I$ )

$j$ : ジョブ番号 ( $j=1,2,\dots,J$ )

$m$ : 装置番号 ( $m=1,2,\dots,M$ )

$t$ : タイムスロット番号 ( $t=1,2,\dots,T$ )

$c_j$ : ジョブ  $j$  の切替コスト

$x_{it}$ : 状態  $i$  のタイムスロット  $t$  での状態量

$X_{Li}$ : 状態  $i$  の状態量下限

$X_{Ui}$ : 状態  $i$  の状態量上限

$\alpha_{ij}$ : 状態  $i$  のジョブ  $j$  によるタイムスロット  $t$  での変化量

$r_{it}$ : 状態  $i$  のタイムスロット  $t$  での出力量

$M_{jm}$ : ジョブ  $j$  で装置  $m$  を使うかどうか

(0: 使わない, 1: 使う)

$B_{ji}$ : ジョブ  $j$  に状態  $i$  が関係あるかどうか

(0: 関係ない, 1: 関係ある)

$\delta_{jt}$ : タイムスロット  $t$  でジョブ  $j$  を実行するかしないか

(0: 実行しない, 1: 実行する)

$l$ : ラグランジュ関数

まず、最適化するコストを装置の切替コストと考える。これを式で表すと目的関数は式(1)のように表せる。

$$\min \sum_{t=1}^T \sum_{j=1}^J c_j (1 - \delta_{j,t-1}) \delta_{jt} \quad (1)$$

このとき、制約条件として状態量の変化、状態量の下限制約、状態量の上制限制約、装置の競合制約(同時利用不可)を式(2)–(5)のようにそれぞれ定義する。

$$\text{subject to } x_{i,t+1} = x_{it} + \sum_{j=1}^J \delta_{jt} \alpha_{ij} - r_{it} \quad \forall i, t \quad (2)$$

$$x_{it} \geq X_{Li} \quad \forall i, t \quad (3)$$

$$x_{it} \leq X_{Ui} \quad \forall i, t \quad (4)$$

$$\sum_j \delta_{jt} M_{jm} \leq 1 \quad \forall t, m \quad (5)$$

次に、制約条件付き最適化問題を、ラグランジュ乗数を用いて制約なし最適化問題に置き換える。これをラグランジュ緩和という。つまり、「制約条件を満たせ」というきつい定式化から「制約条件違反を減らせ」という緩い定式化に変換する。式(1)–(5)において、ラグランジュ乗数  $\lambda$  (装置の競合の制約のラグランジュ乗数)、 $\theta$  (状態量の下限制約のラグランジュ乗数)、 $\mu$  (状態量の上制限制約のラグランジュ乗数)を導入し式(3)、式(4)、式(5)を以下のように緩和する。このとき、 $\lambda$  は装置の使用料も意味する。

$$\begin{aligned} \min \quad l = & \sum_{t=1}^T \sum_{j=1}^J c_j (1 - \delta_{j,t-1}) \delta_{jt} \\ & + \sum_{t=1}^T \sum_{i=1}^I \theta_{it} (X_{Li} - x_{it}) \\ & + \sum_{t=1}^T \sum_{i=1}^I \mu_{it} (x_{it} - X_{Ui}) \\ & + \sum_{t=1}^T \sum_m \lambda_{mt} \left( \sum_{j=1}^J \delta_{jt} M_{jm} - 1 \right) \end{aligned} \quad (6)$$

subject to 式(2)

ここで、式(2)、式(3)、式(4) (ただし式(3)、式(4)では  $\delta$  は陽には表れていない) に関係する決定変数ベクトル  $\delta$  および状態変数ベクトル  $x$  は、ジョブ別に分離可能である。よって、式(2)、式(6)で表される問題の最小化は、ジョブ  $j$  に関する式(7)、式(8)で表される部分問題をそれぞれ独立に最小化することと同じであると考えられることができる。

$$\min \quad l_j = \sum_{t=1}^T c_j (1 - \delta_{j,t-1}) \delta_{jt}$$

$$\begin{aligned}
& + \sum_{i=1}^T \sum_{j=1}^I \theta_{ij} (X_{Li} - x_{it}) B_{ji} \\
& + \sum_{i=1}^T \sum_{j=1}^I \mu_{ij} (x_{it} - X_{Ui}) B_{ji} \\
& + \sum_{i=1}^T \sum_{m=1}^M \lambda_{im} \delta_{ji} M_{jm}
\end{aligned} \tag{7}$$

$$\text{subject to } x_{it+1} = x_{it} + \delta_{ji} \alpha_{ij} - r_{it} \quad \forall i, t \tag{8}$$

式(7)、式(8)は、個々のジョブのスケジュール問題を表しているが、その目的関数には、状態(タンク内の物質など)の干渉に関するペナルティや装置の競合に関するペナルティが存在するため、一方のスケジュール結果が他方のスケジュールに影響を及ぼす関係にある。

ここで、式(7)で  $B_{ji}$  項を導入し、陽に表れない項をジョブ別に分離したことが本定式化の工夫である。分離された後の個々のジョブの意思決定においては、ジョブの実行により直接影響を受ける状態のみを考慮したほうが良い計算結果を得られたためである。この  $B_{ji}$  の導入前は過干渉により良い計算結果が得られなかった。

## 2.2 ラグランジュ乗数の決定法

部分問題を繰り返し解きながら、個々のスケジュールが全体として協調した状態へ徐々に近づいていくように計算を行なう。このときラグランジュ乗数をうまく決定すれば、ランダムな探索を行うより効率よく探索できると考えられる。ここでは、競売の考え方<sup>4)</sup>を利用してラグランジュ乗数を決定する。あるタイムスロットにおいて装置の競合が発生した場合、価格をうまく設定すると、競合したジョブは他のタイムスロットに逃げ、高い使用料を払ってもその装置を使用したいジョブが1つだけ残る。つまり、それぞれのジョブが利己的に振舞っても、装置の価格を調整することで、競合を解消した良いスケジュールが発見できると考える。この方法をマルチエージェント・スケジューリングと名付ける。ここで利己的とは、個々のジョブ(部分問題)のラグランジュ関数を最小にすることである。この価格の調整には劣勾配を用いて値上げの方向性を決める。この問題の劣勾配、 $\lambda$  は装置の不足台数、 $\theta$  は状態量下限の制約違反量、 $\mu$  は状態量上限の制約違反量を意味する。

次に、式(11)に示す双対ギャップを用いて、作成されたスケジュールを評価する。双対ギャップは、式(9)の主問題と式(10)の双対問題のラグランジュ関数の差であり、設定した価格と現実との乖離(価格設定の甘さ)を意味する。この計算では双対ギャップが閾値以下もしくは劣勾配が0になった場合にイテレーションを終了する。この計算手順を Fig.3 に示す。このような手順により状態の干渉および装置の競合を解消しようとする。

$$\text{主問題: } l_l = \min_x l(x, \lambda, \theta, \mu) \tag{9}$$

$$\text{双対問題: } l_u = \max_{\lambda, \theta, \mu} \min_x l(x, \lambda, \theta, \mu) \tag{10}$$

$$\text{双対ギャップ: } l_u - l_l \tag{11}$$

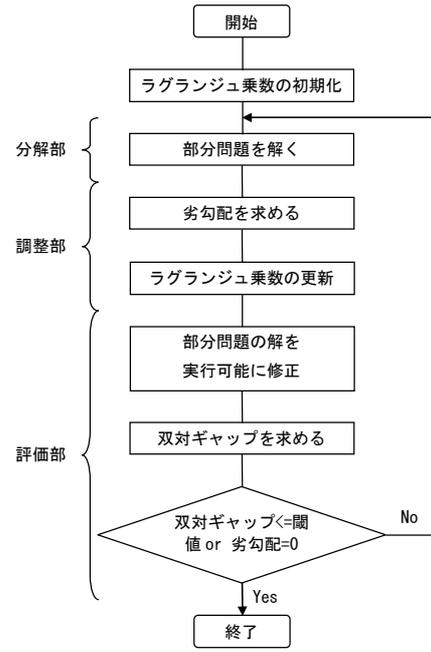


Fig.3 Procedure of Lagrange decomposition and configuration

Step 1: ラグランジュ乗数の初期化  
 $\mu=3, \theta=3, \lambda=0$  を設定する。ジョブ番号  $j=0$  を設定する

Step 2: 部分問題を解く  
 $j \leftarrow j+1$  とし、ジョブ  $j$  について式(7)、式(8)からなる部分問題を解き、各ジョブのスケジュール  $\delta_{ji}$  を求める。  
 $j$  がジョブ数  $J$  を超えた場合は 1 に戻す。

Step 3: 劣勾配を求める  
 $\lambda, \theta, \mu$  の劣勾配

$$\text{subgrad}(\lambda) = \{\dots, \partial l / \partial \lambda, \dots\}$$

$$\text{subgrad}(\theta) = \{\dots, \partial l / \partial \theta, \dots\}$$

$$\text{subgrad}(\mu) = \{\dots, \partial l / \partial \mu, \dots\} \text{ を求める。}$$

Step 4: ラグランジュ乗数の更新

$$\lambda = \lambda + s_i \cdot \text{subgrad}(\lambda) \quad s_i: \text{ステップ幅}$$

さらに  $\mu, \theta$  には、劣勾配だけではなく、制約違反がない場合( $\theta$  が  $X_{Li} < x_{it} < X_0$  のとき、 $\mu$  が  $X_0 < x_{it} < X_{Ui}$  のとき)にも状態量の変化に応じた勾配をつける。

$$\theta_{it} = \begin{cases} \theta_{it} + \max(X_{Li} - x_{it}, 0) & x_{it} < X_{Li} \\ (X_0 - x_{it}) / (X_0 - X_{Li}) & X_{Li} < x_{it} < X_0 \\ 0 & \text{other} \end{cases}$$

$$\mu_{it} = \begin{cases} \mu_{it} + \max(x_{it} - X_{Ui}, 0) & X_{Ui} < x_{it} \\ (x_{it} - X_0) / (X_U - X_0) & X_0 < x_{it} < X_{Ui} \\ 0 & \text{other} \end{cases}$$

Step 5: 部分問題の解を実行可能に修正

Step 2 で求めたスケジュールに制約条件違反がある場合、それを実行可能なスケジュールに修正する（本計算での方策は、競合が起こった場合ジョブを1つ後ろにずらす）。

Step 6: 双対ギャップを求める

Step 2 と Step 5 で求めたスケジュールについてそれぞれラグランジュ関数  $l_l, l_u$  を計算して双対ギャップ  $l_u - l_l$  を求める。

Step 7: 終了条件

1 または 2 のいずれかが成り立つ場合

1. 双対ギャップが閾値以下になる
2. 劣勾配が 0 になってラグランジュ乗数が収束する

ここで、Step 4 の制約違反がない場合にも勾配をつけた点が本アルゴリズムで最も工夫したところである。組み合わせ問題の最適化（離散最適化）では、連続関数の最適化と違って、制約違反が突然発生する場合があります、非常に厄介な問題となる。そこで、制約違反がない場合にも勾配をつけ、制約違反の前兆を知らせる工夫を行った。

### 2.3 動的スケジューリングへの適用の可能性

動的スケジューリングでは、動的問題をあるタイムスロットにおいてデータが一時的に固定された擬似静的問題とみなし、新しい擬似静的問題が定義されるたびに先に示した Fig.3 の計算を行なう<sup>6)</sup>。この場合、予定されていない変化が生じると、擬似静的問題が定義しなおされ、直ちに再計算が行なわれる。ただし、変化は必ずタイムスロットとそれに続くタイムスロットの間で生じるものとする。これを図示すると Fig.4 になる。

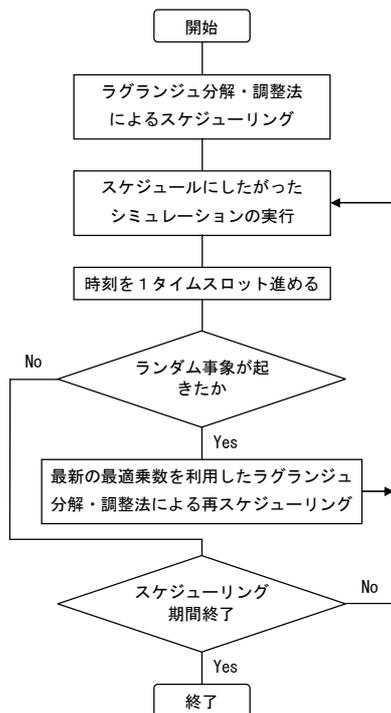


Fig.4 Procedure of dynamic scheduling<sup>6)</sup>

次に、ラグランジュ乗数と収束までの時間について論じる。ラグランジュ分解・調整法において、ラグランジュ乗数（本段落でのみ  $\lambda$  と置く）と最適化に要する時間には密接な関係があり、 $\lambda$  は繰り返しにより最終的に  $\lambda^*$  に接近していく。その後は繰り返し回数が少なくなり計算時間が大幅に削減できる。仮にデータの変化が生じたとき最新の  $\lambda^*$  を初期値として再最適化を行えば、前の  $\lambda^*$  によって求められたスケジュールを受け継ぎ、新しいスケジュールを効率よく探索できると言われている<sup>4)</sup>。このため動的スケジューリングに対する適合性が高いと考えられる。

### 3. 計算例

先に示したスケジューリング・アルゴリズムを ALS スケジューラに実装する。ここでは、実際の開発を前に、本アルゴリズムを CEEF 気体循環システムの植物栽培モジュール O<sub>2</sub> 分離装置のスケジューリング問題に適用し、その性能についてシミュレーションにより検討した。シミュレーションは、MS-Excel の表計算機能と VBA プログラムを用いて行われた。

シミュレーションで用いた CEEF 気体循環システムを Fig.5 に示す。このシステムは動物飼育・居住モジュール、A,B,C,F の 4 つの植物栽培モジュール、O<sub>2</sub> タンク、CO<sub>2</sub> タンク、廃棄物処理装置、CO<sub>2</sub> 分離装置 H、CO<sub>2</sub> 輸送装置、CO<sub>2</sub> 分離装置 P、O<sub>2</sub> 分離装置、O<sub>2</sub> 輸送装置からなる。ただし、O<sub>2</sub> タンク、CO<sub>2</sub> タンクの構成は複数あるタンクを 1 つにまとめてある。このうち O<sub>2</sub> 循環に注目し、O<sub>2</sub> 分離装置のスケジュールを求めた。

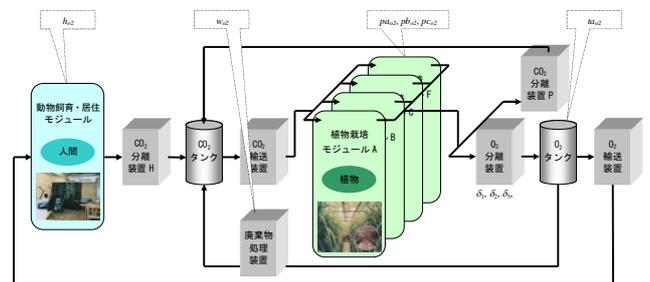


Fig.5 CEEF gas circulation system<sup>2)</sup>

Table 1 Specifications and environmental conditions of modules<sup>1)</sup>

動物飼育・居住モジュール	容積	340m <sup>3</sup> (居住区、動物飼育区、連絡通路)
	O <sub>2</sub> 濃度	Target 20.3%, High 23.5% Low 19.5%
	CO <sub>2</sub> 濃度	High 5000 μLL <sup>-1</sup> 以下
植物栽培モジュール	容積	146.3 m <sup>3</sup> (A,B,C) 239 m <sup>3</sup> (F) 332.2 m <sup>3</sup> (作業室)
	O <sub>2</sub> 濃度	Target 20.3% High 23.5% Low 19.5%
	CO <sub>2</sub> 濃度	明期 700±70μLL <sup>-1</sup> 暗期 1500 μLL <sup>-1</sup> 以下

モジュールの仕様と環境条件を Table 1 に示す。動物飼育・居住モジュールの容積は 340m<sup>3</sup>、その環境は O<sub>2</sub> 濃度が Target 20.3%、

High 23.5%、Low 19.5%、CO<sub>2</sub>濃度が High 5000 μLL<sup>-1</sup>以下である。植物栽培モジュールの容積はA,B,Cのそれぞれが 146.3 m<sup>3</sup>, Fが 239 m<sup>3</sup>、作業室が 332.2 m<sup>3</sup>、その環境は O<sub>2</sub>濃度が Target 20.3%、High 23.5%、Low 19.5%、CO<sub>2</sub>濃度が明期 700±70μLL<sup>-1</sup>、暗期 1500 μLL<sup>-1</sup>以下である。

式(7)、式(8)を用いて O<sub>2</sub>分離装置のスケジューリング問題を定式化すると、計画期間 1 日(タイムスロットを 1 時間とする)の場合、時間 T=24、状態数 I=5、ジョブ数 J=3 (本計算では植物栽培モジュール F を利用していないため)、装置数 M=1 で、変数 72、制約条件数 120 の問題となる。式(8)に相当する状態量の変化に関する制約条件式は式(12)–(17)のように表せる。

$$h_{O_2}(t+1) = h_{O_2}(t) + dO_2(t) \cdot \delta_{ji} - Ch_{O_2} \quad (12)$$

$$pa_{O_2}(t+1) = pa_{O_2}(t) + Cpa_{O_2} - SeO_2 \cdot \delta_{ji} \quad (13)$$

$$pb_{O_2}(t+1) = pb_{O_2}(t) + Cpb_{O_2} - SeO_2 \cdot \delta_{ji} \quad (14)$$

$$pc_{O_2}(t+1) = pc_{O_2}(t) + Cpc_{O_2} - SeO_2 \cdot \delta_{ji} \quad (15)$$

$$ta_{O_2}(t+1) = ta_{O_2}(t) + SeO_2 \cdot \delta_{ji} - dO_2(t) \cdot \delta_{ji} - Cw_{O_2} \cdot \delta_{ji} \quad (16)$$

$$w_{O_2}(t) = \begin{cases} Cw_{O_2} \cdot \delta_{ji} & (t = T_w) \\ 0 & (other) \end{cases} \quad (17)$$

このとき、記号を次のように定義する。

$dO_2$  : O<sub>2</sub>供給量

$Ch_{O_2}$  : 人間の呼吸による O<sub>2</sub>発生量

$Cpa_{O_2}, Cpb_{O_2}, Cpc_{O_2}$  : 植物栽培モジュール a, b, c の植物の光合成による O<sub>2</sub>発生量

$SeO_2$  : O<sub>2</sub>分離装置の O<sub>2</sub>分離量

$Cw_{O_2}$  : 廃棄物処理装置への O<sub>2</sub>供給量

$h(t), pa(t), pb(t), pc(t), ta(t), w(t)$  : 動物飼育・居住モジュール、植物栽培モジュール a, b, c、タンク、廃棄物処理装置のそれぞれの O<sub>2</sub>量

$T_w$  : 廃棄物処理ジョブの開始タイムスロット

$\delta_{ji} \in \{0,1\}$

このときのシミュレーション設定を Table 2 に示す。居住区で 2 人(就寝は 22 時から 6 時まで)の人間が生活し、イネとダイズを栽培して食料を生産する。植物栽培モジュール A でイネ(明期は 0 時から 14 時まで)、B でイネ(明期は 4 時から 18 時まで)、C でダイズ(明期は 8 時から 22 時まで)を栽培し、F では何も栽培しない。また廃棄物処理装置は 8 時に始動するとした。

この設定に基づいて、Excel の表計算機能と VBA プログラムを利用して O<sub>2</sub>分離装置の 1 日間のスケジューリングを行った結果を Fig.6 に示す。(a)はラグランジュ関数の変化、(b)は O<sub>2</sub>分離装置のガントチャート、(c)は植物栽培モジュールの O<sub>2</sub>濃度の変動、(d)は O<sub>2</sub>タンクの状態量の変動を示す。(a)において、 $la, lb, lc$  は植物栽培モジュール PCMA, PCMB, PCMC をそれぞれ処理するジョブのラグランジュ関数値を示す。廃棄物処理のために 8 時に O<sub>2</sub>タンクから廃棄物処理装置へ大量の O<sub>2</sub>が供給され (Fig.6 (d))、O<sub>2</sub>タンクでペナルティが発生したため  $la, lb, lc$  の値が上昇している。その後は

Fig.3 の手順により O<sub>2</sub>分離装置へジョブの割り付けを実行している。そのときの O<sub>2</sub>濃度の変動が(c)でありそれぞれのモジュールの O<sub>2</sub>濃度は許容範囲内(19.5%~23.5%)に制御されている。また、そのときの O<sub>2</sub>分離装置のガントチャートが(c)である。A, B, C は O<sub>2</sub>分離装置が接続している植物栽培モジュールを示している。最後に、O<sub>2</sub>タンクの O<sub>2</sub>量の変動を示した(d)から O<sub>2</sub>量も許容範囲内(0g~10000g)に制御されていることがわかる。

Table 2 Setup values for the simulation

人間	2人, CO <sub>2</sub> : 1402.6 g/day, O <sub>2</sub> : 1077.4 g/day 就寝は 22 時から 6 時まで, 代謝は活動時の 2/3 になる
植物	植物栽培モジュール A, B イネ 442.0g/day 明期(14h) CO <sub>2</sub> : 1884.1 g/day, O <sub>2</sub> : 1454.4 g/day 暗期(10h) CO <sub>2</sub> : 198.7 g/day, O <sub>2</sub> : 164.5 g/day 植物栽培モジュール C ダイズ 194.0g/day 明期(14h) CO <sub>2</sub> : 992.7 g/day, O <sub>2</sub> : 897.0 g/day 暗期(10h) CO <sub>2</sub> : 118.0 g/day, O <sub>2</sub> : 114.7 g/day 植物栽培モジュール F 栽培なし
タンク	CO <sub>2</sub> タンク :5000g, O <sub>2</sub> タンク :5000g 動物飼育・居住モジュール O <sub>2</sub> :84550g, CO <sub>2</sub> :125g 植物栽培モジュール A, B, C O <sub>2</sub> :36435g, CO <sub>2</sub> :125g
状態量 制約	CO <sub>2</sub> タンク : Min 0, Max 10000 g O <sub>2</sub> タンク : Min 0, Max 10000 g 動物飼育・居住モジュール O <sub>2</sub> : Min 81218 g, Max 97878 g, CO <sub>2</sub> : Min 0 g, Max 2083 g 植物栽培モジュール A,B,C O <sub>2</sub> : Min 34947 g, Max 42116 g, CO <sub>2</sub> : Min 0 g, Max 896 g
処理能力 制約	CO <sub>2</sub> 分離装置 H: 58.4 g/h CO <sub>2</sub> 分離装置 P: 21.5 g/h O <sub>2</sub> 分離装置 : 423 g/8h CO <sub>2</sub> 輸送装置 : 942.1 g/12h O <sub>2</sub> 輸送装置 : 44.9 g/h

\* 気体は標準大気での質量に変換されている

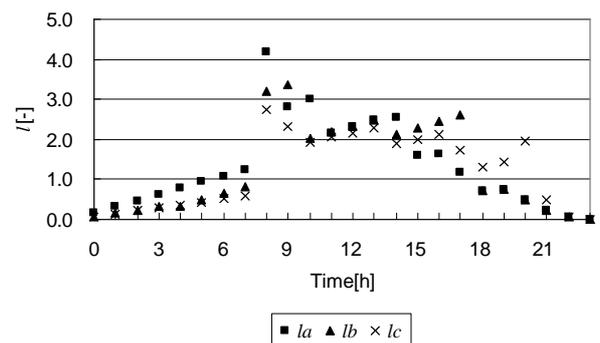


Fig. 6 (a) Values of Lagrange functions

PCM/Time	6	12	18	24
A				
B				
C				

Fig. 6 (b) Gantt chart of O<sub>2</sub> separator

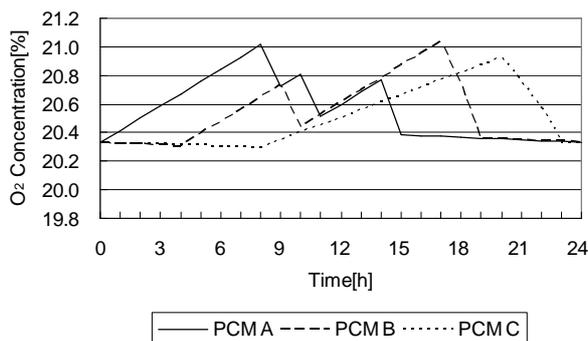


Fig.6 (c) Change of O<sub>2</sub> concentration in PCMs

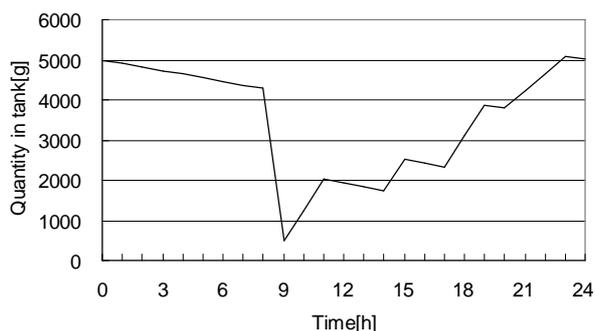


Fig. 6 (d) Change of quantity in O<sub>2</sub> tank

#### 4. まとめ

本論文では、ラグランジュ分解・調整法による定式化、ラグランジュ乗数の決定法、動的スケジューリングへの適用の可能性の3点について議論し、以下のような結論を得た。

##### ラグランジュ分解・調整法による定式化

ALS のスケジューリング問題は、ジョブに対して部分問題への分解が可能である。よって、ラグランジュ分解・調整法を用いて指数的な計算量の増加なしでスケジューリングすることができる。本定式化では、部分問題への分解において  $B_j$  項を導入し、陽に表れない状態の項をジョブ別に分離し、個々のジョブの意思決定で状態間の過干渉を防ぐ工夫を行った。

##### ラグランジュ乗数の決定法

競売のコンセプトにより価格を調整することで、部分問題のスケジューリングの協調を実現することができる。本アルゴリズムでは、

状態量の下限制約と上限制約のラグランジュ乗数の決定において、制約違反がない場合にも勾配をつけ、制約違反の前兆を知らせる工夫を行った。

##### 動的スケジューリングへの適用の可能性

計算量とラグランジュ乗数の引継ぎ特性の点から動的スケジューリングへ利用できる可能性がある。

今回の計算では、3つのジョブからなる例題を用いて開発したアルゴリズムの性能を確認した。本アルゴリズムは、ジョブ数が増えた場合でも計算量の指数的な増加が発生しないため大規模なシステムや動的なスケジューリングに利用できる可能性を持っている。今後、ジョブ数が増えた場合や動的スケジューリングに適用した場合の性能の検証を行う予定である。

#### 参考文献

1. Tako Y., Tani T., Arai R. and Nitta K., Estimation of Flows of Carbon and Oxygen in the CEEF System Based on Data Collected in a Stable Phase of Sequential Crop Cultivation Lasting More than 100 days, SAE Technical Paper Series 2005-01-3108, 2005.
2. Miyajima H., Abe K., Hiroaki T., and Ishikawa Y., Design of Intelligent Control Software for Mini-Earth, 2006-01-2123, SAE Technical Paper Series, 2006.
3. 黒田充, 村松健児, 生産スケジューリング, 朝倉書店, 2002.
4. Zhang Y., Luh P. B., Yoneda K., Kano T. and Kyoya Y., Mixed-model Assembly Line Scheduling Using the Lagrangian Relaxation Technique, IIE Transactions 32(2), pp.125-134, 2000
5. Luh P. B., Hoitmt D. J. Max E. and Pattmati K. R., Schedule generation and reconfiguration for parallel machines, IEEE transactions on Robotics and Automation, Vol.6, No.6, pp. 687-696, 1990.
6. Kuroda M. and Enomoto M., Usefulness of Lagrangian relaxation approaches to production scheduling under a dynamic environment, JAPAN-USA Symposium on Flexible Automation, Vol.2, pp.899-905, 1998.