2007-01-3025

Development of Advanced Life Support Systems Control Software Considering Computational Effort and Mathematical Validity

Hiroyuki Miyajima
Tokyo Jogakkan College

Koichi Abe
Institute for Environmental Sciences

Tomofumi Hirosaki
Space Systems Development Corporation

Yoshio Ishikawa
Nihon University

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.

INTRODUCTION

The Closed Ecology Experiment Facilities (CEEF) was constructed to study propagation and accumulation of ^{14}C released from a reprocessing site of spent nuclear fuel. A habitation experiment using the CEEF was started in 2005. In the future, stays will be gradually extended [1]. In the previous report [2], as shown in Fig. 1, we have been developing the three layered control software for a Control Computer System (CCS) of the CEEF in order to back up the habitation experiments. In this paper, we will show the development of an operation scheduling system (hereinafter referred to as Advanced Life Support system (ALS) scheduler) on one of three layers, the planning and scheduling level, and discuss the development of a scheduling algorithm which does not cause the complexity of the ALS scheduler to be exponentially increased.

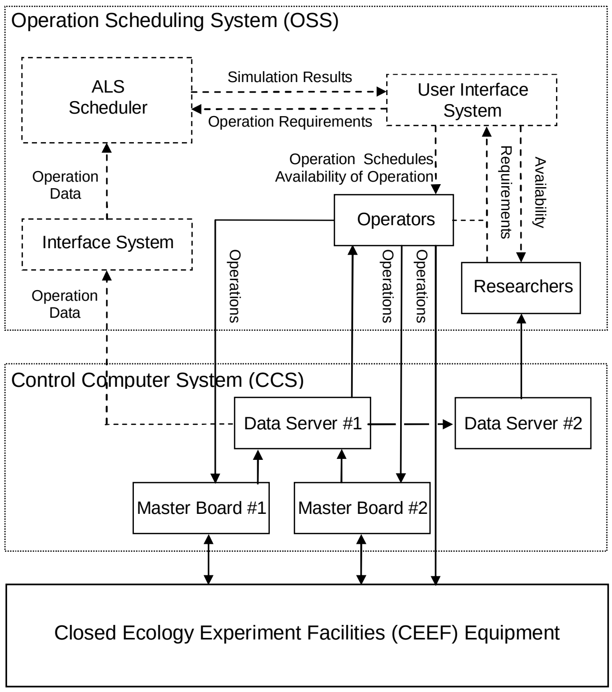

Fig. 1 Expanded CCS of the CEEF

The ALS scheduler will be developed using the MS Visual C++, and is expected to be software that is represented as shown in Fig. 2. The ALS scheduler has the functions of describing a scheduling model using a Planning and Scheduling Language (PSL) [3]; allocating jobs using a scheduling algorithm; and displaying the result as a Gantt chart and a graph for a change of state quantity. Before starting an experiment, operators generate an operation schedule based on the design of the experiments using the scheduler, and confirm the results. When altering the schedule is required or when any abnormality occurs in the facilities after starting the experiment, the operators regenerate the operation schedule based on the operation data, and then confirm the results. The operators manually implement the generated operation schedule.

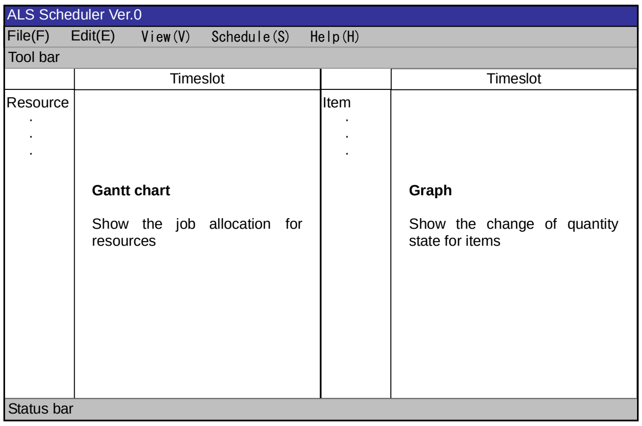

Fig. 2 Screen of the ALS scheduler

A scheduling problem is a combinatorial problem that is complexity is exponentially increased depending on the scale of the problem (Bellman's curse of dimensionality). This problem belongs to the class of NP-complete problems or NP-hard problems. In conventional scheduling, mathematical programming methods (Enumeration method, Dynamic Programming, Branch and Bound method, etc.), meta-heuristics (Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), heuristics inherent to a problem), dispatching rules have been employed. Mathematical programming has difficulty in solving a high dimension problem; and although the meta-heuristics can give a quasi-optimal solution to a certain kind of problem within an adequate period of time, it is necessary to devise a formulation and/or to adjust parameters for each problem. The dispatching rule is a method, which has so far been most frequently used, and is also a solid scheduling, that has no unnecessary calculations, which is applicable to any kind of problem. However, acquiring the rule requires someone with a great deal of experience.

Development of the ALS scheduler aims at the implementation of an algorithm of dynamic scheduling that is capable of handling large-scale problems. The dynamic scheduling designates a problem in which not all data have been given at the time of creating a schedule, or changed after the completion of scheduling. The application of an optimization technique to such a dynamic scheduling has not yet been realistic in terms of complexity and computer capability. Accordingly, in order to apply an optimization technique to the dynamic scheduling, it is necessary to overcome an exponentially increasing complexity in solving a scheduling problem and thereby search a solution within a practically acceptable period. Thus, the dynamic scheduling for a large-scale system becomes possible.

Therefore, the object of this research is to develop a scheduling algorithm that prevents the complexity from exponentially increasing. In the subsequent sections, we will discuss the following four subjects: the algorithm of combinatorial problems and complexity, the application of the Lagrange decomposition and configuration, a method of determining Lagrange multipliers, and the possibility of applying the Lagrange decomposition and configuration to dynamic scheduling.

SCHEDULING ALGORITMS

In Table 1, the orders of the complexity of various algorithms of scheduling are shown. The complexity of the scheduling is represented by the number of jobs and a planning period. To obtain the order of a Lagrange decomposition and configuration, the total number of iterations iterated until a Lagrange multiplier is converged is also used.

Table 1 Complexity of algorithms for single machine scheduling problem [4]

Algorithms	Orders
Enumeration method	$o(T^J)$
Dynamic programming	$o(J^2(T-1))$
Branch and bound approach	$< o(T^J)$
Meta-heuristics	$< o(T^J)$
Lagrange decomposition and configuration	$o(JTU)$

o: orders, J: number of jobs (the number of actions), T: planning period (the number of states), U: total number of iterations iterated until a Lagrange multiplier is converged

When using methods other than dynamic programming and the Lagrange decomposition and configuration, the complexity increases exponentially for a planning period. In dynamic programming, this increasing of the complexity is the problem. The Lagrange decomposition and configuration is one of the most powerful methods for scheduling that is capable of overcoming the above problem. In the present method, while checking an exponentially increasing complexity by using the decomposition, an optimal or a quasi-optimal solution can be sought within a practically acceptable period. In the present method, the allocating of certain jobs is performed by selecting the most desirable timeslots (time intervals of scheduling) for the individual jobs, while neglecting the allocating of the other jobs. When a job competition occurs, balancing is performed to resolve the competition. The process of this balancing is analogous to the methods of loading and leveling. That is, the above process can be considered equivalent to a process in which loading and leveling to be performed based on human judgment are mathematically performed.

Next, a scheduling problem of the ALS is formulated by using the Lagrange decomposition and configuration.

FORMULATION OF LAGRANGE DECOMPOSITION AND CONFIGURATION

Symbols and subscripts are defined as follows:

i: state number (i=1, 2, …, I)

j: job number (j=1, 2, …, J)

m: device number (m=1, 2, …, M)

t: timeslot number (t=1, 2, …, T)

c_j: switching cost of job j

x_{it}: state quantity on timeslot t in state i

X_{Li}: lower bound of state quantity in state i

X_{Ui}: upper bound of state quantity in state i

α_{ijt}: amount of change on timeslot t due to job j in state i

r_{ij}: amount of output on timeslot t in state j

M_{jm}: index of describing whether to use a device m in job j (i.e., not use if =0; use if =1)

B_{ji}: index of describing whether state i is connected with job j (i.e., not connected if = 0; connected if = 1)

δ_{jt} : index of describing whether to execute job j on timeslot t (i.e., not execute if =0; execute if =1)

l: Lagrange function

First, assuming that a cost to be optimized is a switching cost of a device, then an objective function can be expressed using Eq. (1).

$$min \qquad \sum_{t=1}^{T}\sum_{j=1}^{J} c_j\left(1-\delta_{jt-1}\right)\delta_{jt} \tag{1}$$

As constraint conditions for the above equation, the change of state quantity, the constraint of lower bound of state quantity, the constraint of upper bound of state quantity, and the constraint of a competition on a device (not allowing simultaneous use of a device)

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

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

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

$$\sum_{j}^{J}\delta_{jt}M_{jm} \leq 1 \quad \forall t,m \tag{5}$$

LAGRANGE RELAXATION

Next, an optimization problem with constraints is replaced by one without constraints by using Lagrange multipliers. This is referred to as a Lagrange relaxation. That is, the formulation of an optimization problem is changed from a strict formulation in which constraints must be satisfied to a relaxed formulation in which constraint violations must be reduced. In Eqs. (1) to (5), introducing Lagrange multipliers denoted by λ, θ, and μ where λ is for the constraining of a competition on a device; θ is for the constraining of a lower bound of state quantity; and μ is for the constraining of an upper bound of state quantity, then Eqs. (3) to (5) are relaxed as follows:

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

$subject \quad to$ Eq. (2)

where λ also represents the use fee of a device.

DECOMPOSITION TO PARTIAL PROBLEM

A decision variable vector δ and a state variable vector x related to Eqs. (2) to (4), (in Eqs. (3) and (4), δ is not explicitly expressed) are separated for individual jobs. Hence, minimizing the problem expressed using Eqs. (2) and (6) is equivalent to independently minimizing partial problems which are expressed using Eqs. (7) and (8) related to jobs j.

$$\begin{aligned} min \qquad l_j = & \sum_{t=1}^{T} c_j\left(1-\delta_{jt-1}\right)\delta_{jt} \\ & + \sum_{t=1}^{T}\sum_{i=1}^{I}\theta_{it}\left(X_{Li}-x_{it}\right)B_{ji} \\ & + \sum_{t=1}^{T}\sum_{i=1}^{I}\mu_{it}\left(x_{it}-X_{Ui}\right)B_{ji} \\ & + \sum_{t=1}^{T}\sum_{m}^{M}\lambda_{mt}\delta_{jt}M_{jm} \end{aligned} \tag{7}$$

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

Although Eqs. (7) and (8) represent scheduling problems corresponding to individual jobs, these problems are related to each other so that the result of one scheduling influences another scheduling since there are penalties related to interference of states (a substance in a tank, or the like) and a competition on a device.

A devised point in the present formulation is to introduce the term B_{ji} in Eq. (7) and to separate terms not explicitly expressed for individual jobs. This is because, in decision-making of the individual jobs after decomposition, a good result is obtained by considering only a state that is directly influenced due to the execution of a job. Before introducing the term B_{ji}, good results were not obtained due to excessive interference.

COOPERATION BY LAGRANGE DECOMPOSITION AND CONFIGURATION

Computation is performed so that individual scheduling as a whole gradually comes into cooperation while iteratively solving the partial problems. At this time, if the Lagrange multipliers are suitably determined, more effective searching can be expected than random searching. Here, the Lagrange multipliers are determined using the concept of auction [8]. When a competition occurs on a device on a timeslot, the setting of an appropriate price allows one job that has happened to be on the device to escape to another timeslot, so that only a job necessary to use the device remains even paying a high use fee. That is, even if individual jobs behave egoistically, adjusting the price of the device brings the competition into a resolution, thus enabling a good schedule to be created. This method is termed a multi-agent scheduling. What is meant by egoistically is that the Lagrange multipliers in individual jobs (partial problems) are minimized. For the adjustment of the price, the direction of a price increase is determined using the subgradient method. As the subgradients of this problem, λ represents the number of shortages of devices; θ represents the amount of constraint violation of a lower bound of state quantity; and μ represents the amount of constraint violation of an upper bound of state quantity.

Subsequently, using the duality gap expressed in Eq. (11), a created schedule is evaluated. A duality gap is the difference between Lagrange multipliers of a main problem expressed by Eq. (9) and a dual problem expressed by Eq. (10), and also implicates a discrepancy (poor price setting) between a set price and a price in real life. In this computation, when the duality gap becomes not greater than a threshold or when the subgradient becomes 0, the iteration is terminated. A procedure for this computation is shown in Fig. 3. Following the procedure, the interference of states and competition on devices will be eliminated.

Main problem : $l_l = \min_x l(x,\lambda,\theta,\mu)$ (9)

Dual problem : $l_u = \max_{\lambda,\theta,\mu} \min_x l(x,\lambda,\theta,\mu)$ (10)

Duality gap : $l_u - l_l$ (11)

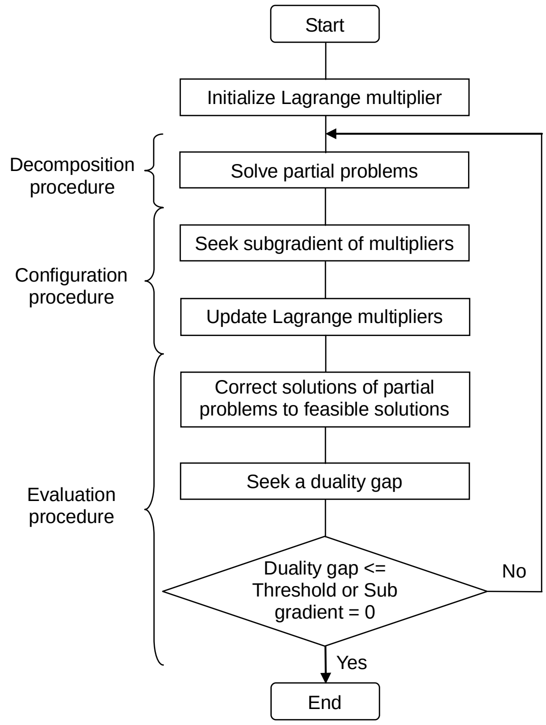

Fig. 3 Lagrange decomposition and configuration

Step 1: Initialize Lagrange multipliers

Set the Lagrange multipliers μ, θ, and λ such that μ =3, θ =3, and λ =0.

Step 2: Solve partial problems

Put $j \leftarrow j+1$; solve partial problems consisting of Eqs. (7) and (8) related to jobs j; and seek schedules δ_{jt} for the individual jobs. When j becomes greater than the job number J, it is returned to 1.

Step 3: Seek subgradients of the Lagrange multipliers.

Seek subgradients of μ, θ, and λ, i.e.,

$$subgrad(\boldsymbol{\lambda}) = \{\cdots, \partial l/\partial\lambda, \cdots\};$$

$$subgrad(\boldsymbol{\theta}) = \{\cdots, \partial l/\partial\theta, \cdots\}; \text{ and}$$

$$subgrad(\boldsymbol{\mu}) = \{\cdots, \partial l/\partial\mu, \cdots\}.$$

Step 4: Update Lagrange multipliers

$\boldsymbol{\lambda} = \boldsymbol{\lambda} + s_i \cdot subgrad(\boldsymbol{\lambda})$ s_i: step width

Furthermore, in addition to the subgradients, gradients depending on the changes of state quantity are provided to μ and θ even when μ and θ do not violate constraints (when θ is on $X_{Li}<x_{it}<X_0$; when μ is on $X_0<x_{it}<X_{Ui}$).

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

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

Step 5: Correct solutions of partial problems to feasible ones

When there is a constraint violation in the schedule obtained in Step 2, the schedule is corrected to a feasible one (a measure taken to this problem in the present computation is that, when a competition occurs, the job is moved back by a timeslot).

Step 6: Seek a duality gap

Compute the Lagrange functions l_l, l_u with respect to the schedules obtained in Steps 2 and 5, and to obtain the duality gap $l_u - l_l$.

Step 7: Conditions of termination

The computation is terminated when one of the following conditions is established.

The duality gap becomes less than or equal to a threshold.

Subgradients become 0, and the Lagrange multipliers converge.

The most devised point in the present algorithm is that gradients are provided to the Lagrange multipliers even when the multipliers do not violate constraints in Step 4. In the discrete optimization of combinatorial problems, constraint violations suddenly occur in many cases unlike the optimization of continuous functions. This is truly a troubling problem. Hence, the present algorithm is so devised that a constraint violation is warned in advance by providing the gradients to the Lagrange multipliers even when there is no constraint violation.

DYNAMIC SCHEDULING

In dynamic scheduling, a dynamic problem is assumed to be a quasi-static problem in which data are temporarily fixed on a certain timeslot, and at each time when a new quasi-static problem is defined, the procedure shown in Fig. 3 is performed. In this case, when an unexpected change occurs, the quasi-static problem is redefined and, immediately, re-computation is performed. Incidentally, it is assumed that a change occurs between a timeslot and another timeslot subsequent thereto. This procedure is shown in Fig. 4 [10].

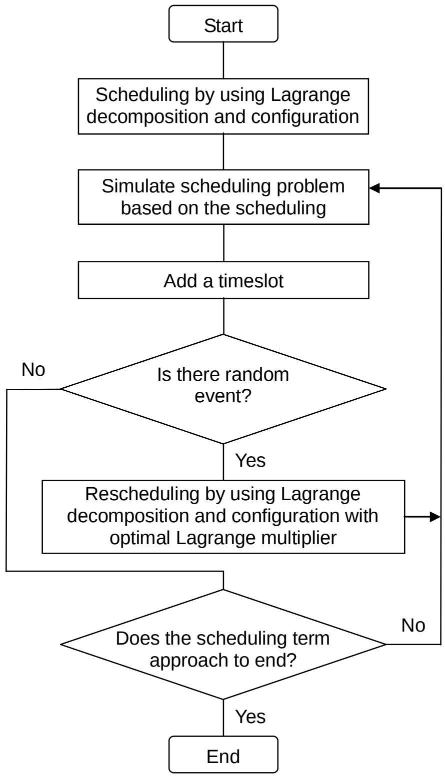

Fig. 4 Procedure of dynamic scheduling

This is discussed below on Lagrange multipliers and convergence time. In the Lagrange decomposition and configuration, there is a close relationship between Lagrange multipliers (represented by λ only in this column) and convergence time required for an optimization, and λ approaches finally to λ* through iterations. Thereafter, the number of iterations is reduced and, consequently, complexity can be reduced to a large extent. If re-optimization is performed with the latest λ* as an initial value when a change in data occurs, a schedule obtained with the previous λ* is inherited so that a new schedule can be effectively searched [4]. Therefore, it has been considered that the degree of consistency to dynamic scheduling is high.

CALCULATION EXAMPLES

The forgoing scheduling algorithm is implemented to the ALS scheduler. Here, before starting full scale development, we executed a simulation where the present method is applied to a scheduling problem of a plant cultivation module O_2 separator of a gas circulation system, and discussed the performance. We carried out the simulation using the spreadsheet and the VBA program of MS-Excel.

Fig. 5 shows the CEEF gas circulation system used in this simulation [2]. This system consists of an Animal and Habitation Module (AHM); 4 Plant Cultivation Modules (PCM) A, B, C, and F; O_2 and CO_2 tanks; O_2 separator; CO_2 separator (H); CO_2 separator (P); O_2 supply unit; CO_2 supply unit; and a solid waste processor. Although the O_2 and CO_2 tanks are expressed as one unit in Fig. 5, there are multiple tanks. The specifications and environmental conditions of modules are shown in Table 2 [1]. The volume of the AHM is 340m^3. O_2 concentration is set as 20.3% (target), 23.5% (high) and 19.5% (low). CO_2 concentration is set as less than 5000 μLL^{-1}. For the PCMs, volumes of A, B, C are each 146.3 m^3, and the volume of F and the preparation room are 239 m^3 and 332.2 m^3. The O_2 concentration is the same as the AHM. CO_2 concentration is set as 700±70μLL^{-1} for light periods and less than 1500□μLL^{-1} for dark periods.

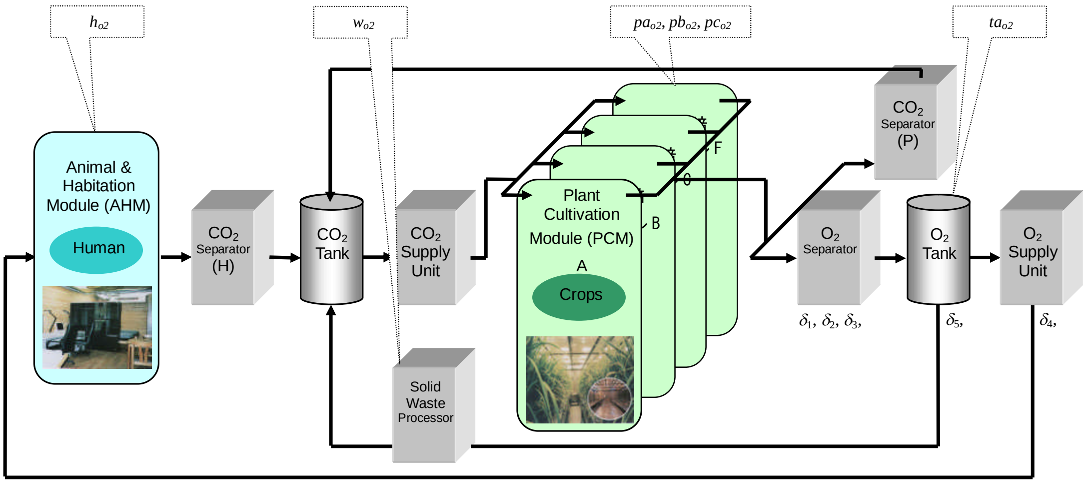

Fig. 5 CEEF gas circulation system

Table 2 Specifications and environmental conditions of modules

AHM	Volume	340m^3 (Habitation Area, Animal Area, Access Aisle)
	O_2 Concentration	Target: 20.3%, High: 23.5% Low: 19.5%
	CO_2 Concentration	High: less than 5000 μLL^{-1}
PCMs	Volume	146.3 m^3 (A,B,C) 239 m^3 (F) 332.2 m^3 (Preparation Room)
	O_2 Concentration	Target 20.3% High 23.5% Low 19.5%
	CO_2 Concentration	Light Period: 700±70μLL^{-1} Dark Period: less than 1500 μLL^{-1}

When formulating a scheduling problem on the O_2 separator using Eqs. (7) and (8), given that a planning period is one day (timeslot is set as one hour), it is formulated with the values given such that T=24, the number of states I=5 (States of PCMs A, B, C, AHM, and O_2 Tank), the number of jobs J=3 (O_2 Separator of PCMs A, B, C is used in the present computation), the number of devices M=1 (O_2 Separator), the number of variables=72 (3 variables δ_1, δ_2, and δ_3 x 24 timeslot), and the number of constraint conditions=120 (5 constraints of Eqs (12) - (16) x 24 timeslot). Equations of constraint conditions, which are related to a change of state quantity corresponding to Eq. (8), are expressed by Eqs. (12) to (17).

$$h_{O2}(t+1) = h_{O2}(t) + dO_2(t)\cdot\delta_{4t} - Ch_{O2} \tag{12}$$

$$pa_{O2}(t+1) = pa_{O2}(t) + Cpa_{O2} - SeO_2\cdot\delta_{1t} \tag{13}$$

$$pb_{O2}(t+1) = pb_{O2}(t) + Cpb_{O2} - SeO_2\cdot\delta_{2t} \tag{14}$$

$$pc_{O2}(t+1) = pc_{O2}(t) + Cpc_{O2} - SeO_2\cdot\delta_{3t} \tag{15}$$

$$\begin{aligned} ta_{O2}(t+1) = ta_{O2}(t) &+ SeO_2\cdot\delta_{1t} + SeO_2\cdot\delta_{2t} + SeO_2\cdot\delta_{3t} \\ &- dO_2(t)\cdot\delta_{4t} - Cw_{O2}\cdot\delta_{5t} \end{aligned} \tag{16}$$

$$w_{O2}(t) = \begin{cases} Cw_{O2}\cdot\delta_{5t} & (t = T_w) \\ 0 & (other) \end{cases} \tag{17}$$

where the symbols used are defined as,

dO_2: amount of supplied O_2;

Ch_{O2}: amount of CO_2 generated by human breathing;

Cpa_{O2}, Cpa_{O2}, and Cpa_{O2}: amount of O_2 generated due to the photosynthesis of plants in PCMs A, B, and C;

SeO_2: amount of O_2 separated from O_2 separator;

Cw_{O2}: amount of O_2 supplied to solid waste processor;

$h(t)$, $pa(t)$, $pb(t)$, $pc(t)$, $ta(t)$, and $w(t)$: respective amounts of O_2 in AHM, PCMs A, B, C, and solid waste processor;

T_w: start timeslot of a solid waste process job; a solid waste processor was assumed to start at 8 o'clock; and

$\delta_{jt} \in \{0,1\}$.

Table 3 Setup values for the simulation

Eco-Nauts	2 people, CO_2 : 1402.6 g/day, O_2 : 1077.4 g/day They sleep from 22 to 6 o'clock, and their metabolism is two thirds that of normal activity while sleeping.
Plants	PCM A and B: Rice (442.0 g/day) Light Period (14h) CO_2: 1884.1 g/day, O_2: 1454.4 g/day Dark Period (10h) CO_2: 198.7 g/day, O_2: 164.5 g/day PCM C: Soybeans (194.0 g/day) Light Period (14h) CO_2: 992.7 g/day, O_2: 897.0 g/day Dark Period (10h) CO_2: 118.0 g/day, O_2: 114.7 g/day PCM F: No Plant
Stocks	CO_2 Tank: 5000 g, O_2 Tank: 5000 g AHM; O_2: 84550 g, CO_2: 125 g PCM A, B, and C; O_2: 36435 g, CO_2: 125 g
Stock Levels	CO_2 Tank: Min 0 g, Max 10000 g O_2 Tank: Min 0 g, Max 10000 g AHM; O_2: Min 81218 g, Max 97878 g, CO_2: Min 0 g, Max 2083 g PCM A, B, and C; O_2: Min 34947 g, Max 42116 g, CO_2: Min 0 g, Max 896 g
Load Levels	CO_2 Separator: 58.4 g/h O_2 Separator: 423 g/8h CO_2 Supply Unit: 942.1 g/12h O_2 Supply Unit: 44.9 g/h

O_2 and CO_2 are expressed in grams in normal atmosphere.

Table 3 shows the setup values for the simulation. Two people (Eco-Nauts) live, cultivating rice and soybeans, to produce their own food for themselves. The scheduling of human activity was given before the scheduling by the ALS Scheduler. The individuals sleep from 22 to 6 o'clock. They cultivate rice in PCM A (the light period is 0 to 14 o'clock) and PCM B (the light period is 4 to 18 o'clock), and soybeans in PCM C (the light period is 8 to 22 o'clock). They do not cultivate in PCM F. The values of stocks (Stocks), stock constraints (Stock Levels), and load constraints (Load Levels) of the items are shown in Table 3.

In Figs. 6 (a) to 6 (d), results are shown which were obtained by performing one-day scheduling of O_2 separator based on the above set values using the spreadsheet and the VBA program of Excel. Fig. 6 (a) shows the changes of Lagrange functions; Fig. 6 (b) shows a Gantt chart; Fig. 6 (c) shows the change of O_2 concentration in the PCMs; and Fig. 6 (d) shows the change of quantity in the O_2 tank. In Fig. 6 (a), *la*, *lb* and *lc* denote, respectively, the values of the Lagrange functions of the jobs handling the PCMs A, B, and C. For carrying out solid waste processor, at 8 o'clock, a large amount of O_2 was supplied from the O_2 tank to the solid waste processor (refer to Fig. 6 (d)); and since a penalty occurred in the O_2 tank, the values *la*, *lb* and *lc* increased. Then, allocations of jobs to the O_2 separator were performed in accordance with the procedure shown in Fig. 3. In so doing, the change of O_2 concentration is controlled to that shown in Fig. 6 (c), and the O_2 concentrations of the respective modules are controlled to be in the allowable range of 19.5% to 23.5%. Moreover, a Gantt chart of the O_2 separator is Fig. 6 (c). A, B, and C, respectively, denote the PCMs to which the O_2 separator is connected. Finally, it can be observed from Fig. 6 (d), showing the change of the quantity in the O_2 tank, that the quantity of O_2 is controlled to be in the allowable range of 0g to 10000g.

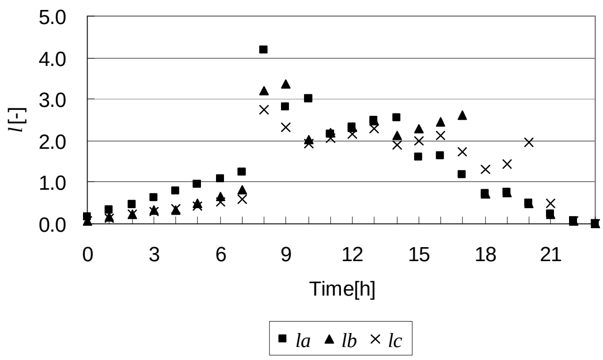

Fig. 6 (a) Values of Lagrange functions

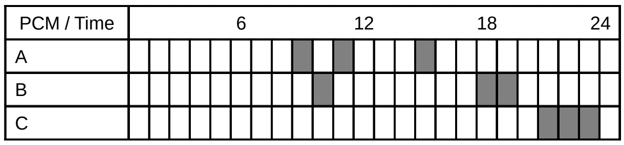

Fig. 6 (b) Gantt chart of O_2 separator

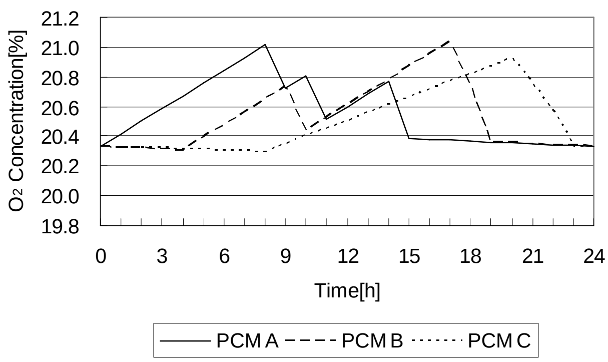

Fig.6 (c) Change of O_2 concentration in PCMs

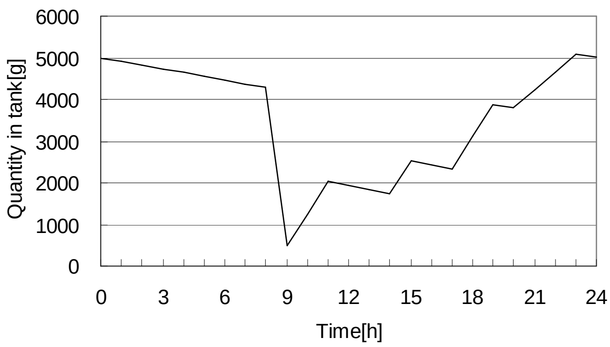

Fig. 6 (d) Change of quantity in O_2 tank

In view of the complexity and inheriting property of Lagrange multipliers, there is the possibility of an application to dynamic scheduling.

In the present computation, the performance of the developed algorithms using the example that consisted of three jobs was validated. Since the present algorithm does not cause the complexity to be exponentially increased even when the number of jobs is increased, there is a possibility that the algorithm is applicable to large-scale systems and dynamic scheduling. In the future, we will validate the performance of the present algorithm when increasing the number of jobs and applying to dynamic scheduling.

CONCLUSION

In this paper, we discussed the following four subjects: the algorithm of combinatorial problems and complexity, the application of the Lagrange decomposition and configuration, a method of determining Lagrange multipliers, and the possibility of applying the Lagrange decomposition and configuration to the dynamic scheduling; and reached the following conclusions.

<u>Algorithm of combinatorial problems and complexity</u>

The Lagrange decomposition and configuration is capable of scheduling to a problem, which can be decomposed into partial problems, in a small complexity compared with other methods.

<u>Application of Lagrange Decomposition and Configuration</u>

For a given job, an ALS scheduling problem can be decomposed into partial problems. Hence, using the Lagrange decomposition and configuration, scheduling can be performed without an exponentially increasing complexity. The present formulation is devised in such a way that the term B_{ji} is introduced and terms of state, not explicitly expressed, are separated for individual jobs, so that occurrence of excessive interference between states in decision-making of individual jobs is prevented.

<u>Method of determination of Lagrange multipliers</u>

Cooperation for the scheduling of partial problems can be achieved by adjusting the price based on the concept of auction. For the determination of the Lagrange multipliers of the lower and upper bound constraints, the present formulation is devised so that gradients are provided to the Lagrange multipliers even when there is no constraint violation and, thereby, warning of an occurrence of a constraint violation is in advance.

<u>Possibility of an application to dynamic scheduling</u>

ACKNOWLEDGMENTS

This report includes part of the results from research conducted under contract with Aomori Prefectural Government, Japan.

REFERENCES

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., Hirosaki T., and Ishikawa Y., Design of Intelligent Control Software for Mini-Earth, 2006-01-2123, SAE Technical Paper Series, 2006.
3. http://www.img.k.hosei.ac.jp/pslib/Default.htm
4. Kuroda M, and Muramatsu K., Production Scheduling, Asakura Publishing, 2002 (Japanese).
5. Sutton R. and Barto A., Reinforcement Learning: An Introduction: The MIT Press, 1998.
6. Sadiq S. and Habib Y., Interactive Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems, IEEE, 2002.
7. Hromkovic J., Algorithmics for Hard Problems, Springer, 2001.
8. 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
9. 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.
10. 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.

CONTACT

Hiroyuki Miyajima
Tokyo Jogakkan College
1105 Tsuruma, Machida-shi, Tokyo 194-0004 Japan
E-mail: miyajima@m.tjk.ac.jp

DEFINITIONS, ACRONYMS, ABBREVIATIONS

AHM: Animal and Habitation Module

ALS: Advanced Life Support systems

CCS: Control Computer System

CEEF: Closed Ecology Experiment Facilities

OSS: Operation Scheduling system

PCM: Plant Cultivation Module

PSL: Planning and Scheduling Language