Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

# An Integrated Column Generation and Lagrangian Relaxation for Flowshop Scheduling Problems

Tatsushi Nishi, Yukinori Isoya, Masahiro Inuiguchi
Mathematical Science for Social Systems
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama-cho, Toyonaka city, 560-8531, Japan
Email: nishi@sys.es.osaka-u.ac.jp

***Abstract*—In this paper, we address a new integration of column generation and Lagrangian relaxation for solving flowshop scheduling problems to minimize the total weighted tardiness. In the proposed method, initial columns are generated by using near-optimal dual solution using the Lagrange multipliers derived by Lagrangian relaxation method. After the generation of base columns, the column generation is executed. Computational results demonstrate that the integrated column generation and Lagrangian relaxation can drastically speed up the conventional column generation.**

***Keywords*—scheduling, column generation, Lagrangian relaxation, flowshop**

## I. Introduction

Flowshop scheduling problem to minimize the total weighted tardiness is known to be NP-hard combinatorial optimization problem. In the flowshop problem, there is a set of jobs that have to be processed on the multi-stage flowshop where each stage is composed of identical machine. Each machine can process one operation at a time where a job consists of a set of operations for plural stages. These operations have to be processed sequentially satisfying precedence constraints between the stages. Each operation has a fixed processing time where the preemption and splitting are not allowed. The problem to minimize the total tardiness is known to be NP-hard in the ordinary sense even for single machine, and NP-hard in the strong sense. Exact algorithms, heuristic algorithms have been extensively studied. A well-known heuristic is NEH algorithm[1]. A branch and bound on the makespan[2][3], tabu search[4], iterated greedy algorithm[5] have been addressed.

Column generation is an effective method for obtaining tighter lower bound for combinatorial optimization problems. Recently, column generation is widely used for railway crew scheduling[6], vehicle routing problem[7], patients scheduling in hospital[8], etc. This is due to the fact that the optimal lower bound for continuous relaxation of set partitioning formulation can be ensured for column generation. However, it is also well known that convergence of column generation is extremely slow due to degeneration in solving restricted master linear programming problem[9]. Recently, several improvements for column generation have been studied. Akker et al. (2002) developed a combined column generation and Lagrangian relaxation (LR) for single machine scheduling problem with common due date. They reported an improvement of convergence of column generation by using the lower bound obtained by LR as the criterion of convergence[10]. Huisman et al. (2003) presented the reduction of computation time by combining column generation and LR to generate and add columns that can be derived by subgradient optimization in LR algorithm[11].

In this paper we address a new integration of column generation and Lagrangian relaxation to improve the convergence of conventional column generation. The new idea of the proposed method is that the promising base columns are generated by LR method beforehand by using the relationship between Lagrange multipliers and simplex multipliers. Then the column generation is executed. The total computation time is expected to be reduced compared with the conventional column generation.

The paper consists of the following sections. In Section II, column generation for flowshop scheduling problems is explained. In Section III, integrated column generation and Lagrangian relaxation is proposed. The algorithm for generating promising base columns is developed. Computational experiments are shown in Section IV. Section V concludes the paper and mentions our future works.

## II. Column generation for flowshop scheduling problems

### A. Problem formulation

The flowshop problem considered in this paper is to schedule a set of $N$ jobs on single machine with $L$ stages to minimize the total weighted tardiness. Each job requires a set of $L$ operations where $N$ is the number of jobs, $L$ is the number of production stages, and $H$ is the total time horizon to be long enough to complete all of the operation of jobs. The following sets, decision variables and parameters are used for formulations.

**Decision variables:**

$c_{i,l}$: completion time of the operation of job $i$ at stage $l$

**Parameters:**

$d_i$: due date of job $i$

$w_i$: weight of job $i$

$p_{i,l}$: processing time of the operation of job $i$ at stage $l$

Let $a_{i,l,t}$ be a binary variable that takes 1 if the operation of job $i$ at stage $l$ is processed on time $t$, and 0 otherwise.

$$a_{i,l,t} = \begin{cases} 1 & (c_{i,l} - p_{i,l} + 1 \le t \le c_{i,l}) \\ 0 & \text{(otherwise)} \end{cases} \tag{1}$$



978-1-4244-2794-9/09/$25.00 ©2009 IEEE

The flowshop scheduling problem $(P)$ is formulated as follows.

$$(P) \quad \min_{\{c_{i,l}\}} \sum_{i=1}^{N} w_i T_i \tag{2}$$

s. t.

$$T_i = \max\{0,\ c_{i,L} - d_i\}, \quad i = 1, \ldots, N, \tag{3}$$

$$c_{i,l} \geq p_{i,l}, \quad i = 1, \ldots, N, \quad l = 1, \ldots, L, \tag{4}$$

$$c_{i,l-1} \leq c_{i,l} - p_{i,l}, \quad i = 1, \ldots, N, \quad l = 2, \ldots, L, \tag{5}$$

$$\sum_{i=1}^{N} a_{i,l,t} \leq 1, i = 1, \ldots, N, l = 1, \ldots, L, t = 1, \ldots, H. \tag{6}$$

The objective function of (2) is to minimize the total weighted tardiness. Constraints (3) define the tardiness for job $i$. Constraints (4) ensure the starting time constraint from the processing time requirement. Constraints (5) denote the precedence constraints that an operation cannot be started until its preceding operation is finished. Constraints (6) state machine capacity constraints. The overall objective is to minimize the total weighted tardiness subject to the constraints mentioned above. The problem is known to be NP-hard in the strong sense. Therefore, the exact algorithms will be intractable for large-sized problems.

### B. Set partitioning problem reformulation

The original problem can be reformulated as a set partitioning problem by Dantzig-Wolfe decomposition. The following variables are defined for the reformulation.

**Sets:**

$\Omega$: the set of all possible schedule $s$ for each job

**Parameters:**

$c_{s,l}$: completion time for schedule $s$ for each job at stage $l$,

$p_{s,l}$: processing time for schedule $s$ for each job at stage $l$,

$w_s$: weight for schedule $s$ for each job,

$T_s$: tardiness for schedule $s$ for each job.

Let $y_s$ be 0-1 binary variable that takes 1 if schedule $s$ for each job is adopted, and 0 otherwise, $X_s$ be 0-1 binary variable that takes 1 if schedule $s$ for each job is related to job $i$, and 0 otherwise. The flowshop scheduling problem can be reformulated as set partitioning problem.

$$(SP) \quad \min_{\{y_s\}} \sum_{s \in \Omega} w_s T_s y_s \tag{7}$$

$$\text{s. t.} \quad \sum_{s \in \Omega} X_{s,i} y_s = 1, \quad i = 1, \ldots, N \tag{8}$$

$$\sum_{s \in \Omega} a_{s,l,t} y_s \leq 1, \quad l = 1, \ldots, L,\ t = 1, \ldots, H \tag{9}$$

$$y_s \in \{0, 1\}, \quad \forall s \in \Omega \tag{10}$$

### C. Column generation

The restricted master problem of the continuous relaxation $(LSP)$ for $(SP)$ can be formulated as

$$(LRSP) \ \min_{y_s} \sum_{s \in \overline{\Omega}} w_s T_s y_s \tag{11}$$

$$\text{s. t.} \ \sum_{s \in \overline{\Omega}} X_{s,i} y_s = 1, \quad i = 1, \ldots, N \tag{12}$$

$$\sum_{s \in \overline{\Omega}} a_{s,l,t} y_s \leq 1, \quad l = 1, \ldots, L,\ t = 1, \cdots, H \tag{13}$$

$$y_s \geq 0, \quad \forall s \in \overline{\Omega} \tag{14}$$

where $\overline{\Omega}$ is a subset of $\Omega$. A column represents a feasible schedule for one job. In the column generation, the master problem with the restricted number of columns $(LRSP)$ is solved. Let $\pi^*$, $\lambda^*$ be the optimal dual solution for the constraint of (12) and (13), respectively. If there exists a column $s$ that makes the reduced cost

$$\overline{c_s} = w_s T_s - \sum_{i=1}^{N} X_{s,i} \pi_i^* - \sum_{l=1}^{L} \sum_{t=1}^{H} a_{s,t,l} \lambda_{t,l}^* \tag{15}$$

negative, the column is added to the restricted column $\overline{\Omega}$. The value of objective function $z_{LRSP}$ is descreased when $LRSP$ is solved repeatedly. If there exists no column that makes the reduced cost negative, the current solution is optimal for $LSP$.

### D. Pricing problem

The pricing problem is to derive a column that can make the reduced cost negative. The problem can be reformulated as

$$\min_{s} \left( w_s T_s - \sum_{i=1}^{N} X_{s,i} \pi_i^* - \sum_{l=1}^{L} \sum_{t=c_{s,l}-p_{s,l}+1}^{c_{s,l}} \lambda_{t,l}^* \right) \tag{16}$$

$$\text{s. t.} \quad c_{i,l} \geq p_{i,l}, \quad l = 1, \ldots, L \tag{17}$$

$$c_{s,l-1} \leq c_{s,l} - p_{s,l}, \quad l = 2, \ldots, L \tag{18}$$

The pricing problem can be decomposed into each job-level subproblem $(PS_i)$ that can be solved by dynamic programming. The dynamic programming recursion is formulated as

$$f_{i,l}(x) = \begin{cases} \kappa_{i,l}(x) & (l = 1) \\ \kappa_{i,l}(x) + \min\limits_{0 \leq z \leq x - p_{i,l}} f_{i,l-1}(z) & (l \neq 1) \end{cases}$$

$$\text{where } \kappa_{i,l}(x) = \begin{cases} \sum\limits_{t=x-p_{i,l}+1}^{x} \overline{\lambda_{t,l}^*} & (l \neq L) \\ w_i T_i + \sum\limits_{t=x-p_{i,l}+1}^{x} \overline{\lambda_{t,l}^*} & (l = L) \end{cases} \tag{19}$$

and $\overline{\lambda^*} = -\lambda^*$. The dynamic programming recursion computes the optimal completion time for each job so that the sum of the weighted tardiness and the simplex multiplier is minimized. The optimal completion time can be obtained by $c_{i,l} = \arg \min\limits_{0 \leq c_{i,l} \leq c_{i,l+1} - p_{i,l+1}} f_{i,l}$ recursively from stage $L$ to stage 1. The computational complexity of solving $(PS_i)$ is $O(HL)$.

### E. Algorithm of column generation

Step 0: **Generation of initial column**

The initial schedule is generated by iterated greedy algorithm[5]. The derived schedule is set as the initial column.

Step 1: **Solving restricted master problem ($LRSP$)**
The linear programming problem is solved and then the dual solution is derived.

Step 2: **Solving pricing problem**
Each job-level subproblem ($PS_i$) ($i = 1, \ldots, N$) is solved by dynamic programming using the dual solution of $LRSP$. If the solution that makes the reduced cost negative is obtained, the column is added to the restricted master problem.

Step 3: **Evaluation of optimality**
If all of the reduced costs are non-negative, the current solution for $LRSP$ is the optimal solution for $LSP$ and go to Step 4. Otherwise return to Step 1.

Step 4: **Construction of feasible solution**
The solution for $LSP$ is not feasible for $SP$. Therefore the construction of feasible solution is necessary. This procedure is explained in the next section.

### *F. Construction of feasible solution*

To construct a feasible solution from the solution of $LSP$, branch and bound algorithm with the derived columns by column generation is often used. To derive a feasible solution effectively, column rounding heuristic is proposed[8]. The algorithm consists of the following steps.

Step 0: Initialization. $I = \phi$.

Step 1: Select $s$ from the solution of $LRSP$ where $y_s = 1$. If $j_s \notin I$ for the corresponding job $j_s$, $I := I \cup \{j_s\}$. If $|I| = N$ then go to Step 4. Otherwise go to Step 2.

Step 2: Select $s$ where the value of fractional part of $y_s$ is maximum and modify to $y_s = 1$. $I := I \cup \{j_s\}$. If $|I| = N$ then go to Step 4. Otherwise go to Step 3.

Step 3: Execute column generation algorithm with fixed $y_s = 1$ which is determined at Step 1 and Step 2. Then return to Step 1.

Step 4: The current solution is regarded as a feasible solution for $SP$ and the algorithm is completed.

## III. Combined Column Generation and Lagrangian Relaxation

### *A. Lagrangian relaxation*

If the machine capacity constraints is relaxed by Lagrangian multiplier $\mu_{t,l}$, the relaxed problem can be written as

$$(LR) \quad \min_{\{c_{i,l}\}} L(\mu) = \sum_{i=1}^{N} w_i T_i - \sum_{t=1}^{H}\sum_{l=1}^{L}\left(1 - \sum_{i=1}^{N} a_{i,l,t}\right)\mu_{t,l}$$

$$\text{s. t.} \quad c_{i,l} \geq p_{i,l}, \quad i = 1, \ldots, N, \; l = 1, \ldots, L \quad (20)$$

$$c_{i,l-1} \leq c_{i,l} - p_{i,l}, \quad i = 1, \ldots, N, \; l = 2, \ldots, L \quad (21)$$

$$\mu_{t,l} \geq 0, \quad l = 1, \ldots, L, \; t = 1, \ldots, H \quad (22)$$

The relaxed problem can be decomposed into individual job-level subproblem that can be solved by dynamic programming. The dual problem can be formulated as

$$(LD) \quad \max_{\mu} q(\lambda, c) \quad (23)$$

$$q(\lambda, c) = \min_{c \in \mathcal{F}} \sum_{i=1}^{N} w_i T_i - \sum_{t=1}^{H}\sum_{l=1}^{L}\left(1 - \sum_{i=1}^{N} a_{i,l,t}\right)\mu_{t,l} \quad (24)$$

where $\mathcal{F} = \{\, c_{i,l} \mid c_{i,l} \text{ satisfies } (20),\ (21)\}$. The Lagrangian dual problem can be solved by subgradient method. $\gamma$ is subgradient of $q(\lambda, c)$, $\alpha$ is a constant.

$$\gamma_{t,l} = \frac{\partial L(\mu)}{\partial \mu_{t,l}} = \sum_{i=1}^{N} a_{i,l,t} - 1, \quad (25)$$

$$\mu_{t,l} = \max\{0,\ \mu_{t,l} + \alpha \frac{UB - LB}{\sum_{t=1}^{H}\sum_{l=1}^{L}\gamma_{t,l}^2}\gamma_{t,l}\}. \quad (26)$$

### *B. Relation between column generation and Lagrangian relaxation*

It is well known that the Lagrangian dual ($LD$) is equivalent to the dual problem for ($LSP$)[12]. The relation between the Lagrangian relaxation and column generation is shown in Fig. 1. The relation between the Lagrange multipliers and simplex multipliers is represented by using the following equations.

$$\lambda_{t,l} = -\mu_{t,l}, \quad 1 \leq t \leq H, 1 \leq l \leq L \quad (27)$$

$$\pi_i = \min\left(w_i T_i + \sum_{t=1}^{H}\sum_{l=1}^{L} a_{i,l,t}\mu_{t,l}\right), \quad 1 \leq i \leq N \quad (28)$$

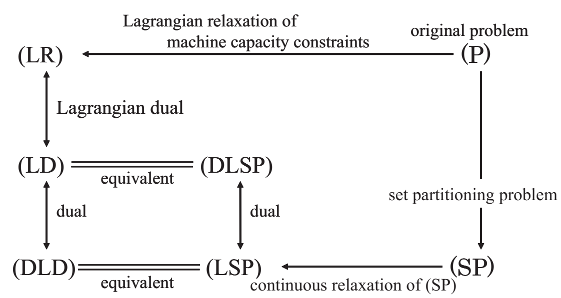


Fig. 1. Relation between column generation and Lagrangian relaxation

## IV. Integration of Column Generation and Lagrangian Relaxation

### *A. Construction of candidates of base columns*

In the column generation algorithm, columns are successively generated and added to the restricted master problem until the optimality condition is satisfied. At the reach of convergence of column generation, optimal solution for $LSP$ can be derived. The derived columns are divided into base columns and non-base columns. If the columns are base columns, the reduced cost is zero, otherwise the reduced cost is non-negative for non-base columns. The necessary condition for being a base column is

$$w_i T_i - \sum_{l=1}^{L} \sum_{c_{i,l}-p_{i,l}+1}^{c_{i,l}} \lambda^*_{t,l} - \pi^*_i = 0 \quad (29)$$

where $\pi^*_i$, $\lambda^*_{t,l}$ is the optimal dual solution for ($LSP$). This condition is not sufficient because there exist non-base column that makes the reduced cost zero. On the other hand, all of reduced costs are non-negative at the reach of convergence for column generation. The following condition holds.

$$\min\left(w_iT_i - \sum_{l=1}^{L}\sum_{t=c_{i,l}-p_{i,l}+1}^{c_{i,l}} \lambda^*_{t,l}\right) = \pi^*_i, i=1,\ldots,N \quad (30)$$

All of base columns can be enumerated when $c_{i,l}$ satisfying (30) can be obtained.

### B. Acceleration of column generation

The base columns are required to obtain an optimal lower bound for column generation. The number of iterations for column generation is only one when the base columns are known. The number of iterations is expected to be reduced by enumerating near-optimal base columns before the column generation. To derive the base columns, near optimal dual solution constructed by Lagrange multipliers derived by Lagrangian relaxation is available. From the viewpoint, we propose a new integration of column generation and Lagrangian relaxation. Let $\mu^*_{t,l}$ be a near-optimal Lagrange multiplier. The near-optimal dual solution $\tilde{\pi}^*_i$, $\tilde{\lambda}^*_{t,l}$ can be derived by

$$\tilde{\lambda}^*_{t,l} = -\mu^*_{t,l}, \quad 1\le t\le H,\ 1\le l\le L \quad (31)$$

$$\tilde{\pi}^*_i = \min\left(w_iT_i + \sum_{t=1}^{H}\sum_{l=1}^{L} a_{i,l,t}\mu^*_{t,l}\right), i=1,\ldots,N \quad (32)$$

By using dual solution (31) and (32), the columns are generated to satisfy

$$\min\left(w_iT_i - \sum_{l=1}^{L}\sum_{t=c_{i,l}-p_{i,l}+1}^{c_{i,l}} \tilde{\lambda}^*_{t,l}\right) = \tilde{\pi}^*_i \quad (33)$$

### C. Algorithm of generation of base columns

In this section, we propose an algorithm to generate base columns to satisfy (33). The computational complexity is $O(H^L)$ if a brute force search algorithm is executed. Therefore we developed an efficient algorithm. The following recursion is computed.

$$f_{i,l}(x) = \begin{cases} -\sum_{t=x-p_{i,l}+1}^{x} \tilde{\lambda}^*_{t,l} & (l=1) \\ -\sum_{t=x-p_{i,l}+1}^{x} \tilde{\lambda}_{t,l} + \min_{0\le z\le x-p_{i,l}} f_{i,l-1}(z) & (l\ne 1) \end{cases}$$

The criteria function for adopting base column is defined as

$$H_i(c_{i,L}) = w_iT_i - \sum_{t=c_{i,L}-p_{i,L}+1}^{c_{i,L}} \tilde{\lambda}_{t,L} + \min_{0\le x\le c_{i,L}-p_{i,L}} f_{i,L-1}(x) \quad (34)$$

The function depends only on the completion time $c_{i,L}$ for job $i$ at last stage. This makes it easier to compute base columns with restricted number of columns. The third term of right-hand side of (34) can be computed by the recursion (34). If the following condition is satisfied, $c_{i,L}$ is fixed. Then the recursion is computed from stage $L-1$ to stage 1.

$$c_{i,L-1} = \arg\min_{0\le x\le c_{i,L}-p_{i,L}} f_{i,L-1}(x) \quad (35)$$

If the completion time for job is computed for all stages, the column is added to a set of base columns. The computational complexity is $O(H^2L)$. The detailed algorithm for the generation of base columns is as follows.

Step 0: Set $i=1$, $x=0$.

Step 1: Compute $h = \min\sum_{l=1}^{L-1}\sum_{c_{i,l}-p_{i,l}+1}^{c_{i,l}} \tilde{\lambda}^*_{t,l}$ by dynamic programming recursion.

Step 2: Compute $H(x) = w_iT_i + \sum_{t=x-p_{i,j}+1}^{x} \tilde{\lambda}^*_{t,l} + h$.

Step 3: If $\tilde{\pi}^*_i - \varepsilon \le H(x) \le \tilde{\pi}^*_i + \varepsilon$ is satisfied, compute $c_{i,l-1},\ldots,c_{i,1}$ by (34) and add to the set of base columns.

Step 4: $x := x+1$. If $h > H$ go to Step 5, otherwise return to Step 1.

Step 5: $i := i+1$. If $i > N$, the algorithm is completed. Otherwise return to Step 1.

### D. Algorithm of the integrated column generation and Lagrangian relaxation

The proposed algorithm consists of the following steps.

Step 0: Execute Lagrangian relaxation

Step 1: Generation of near-optimal dual solution from the solution of Lagrangian relaxation.

Step 2: The initial feasible schedule is created by iterated greedy algorithm.

Step 3: Generation of candidates of base columns.

Step 4: Execute column generation

## V. Computational experiments

Computational experiments are demonstrated to show the effectiveness of the proposed method in this section.

### A. Problem instances

Ten problem instances are created by random numbers on uniform distribution on the interval shown in Table I. $P$ is the lower bound for makespan which is given by

$$P = \max\left\{\max_{1\le l\le L}\left(\sum_{i=1}^{N} p_{i,l} + \min_i \sum_{l'=1}^{l-1} o_{i,l'} + \min_i \sum_{l'=l+1}^{L} p_{i,l'}\right), \max_i \sum_{l=1}^{L} p_{i,l}\right\} \quad (36)$$

The time horizon $H = \max_i d_i + (L-1)\max_{i,l} p_{i,l} + N\max_{i,l} p_{i,l}$ is used[13]. The program is coded by C++ language of Visual C++ 2008 Express Edition. CPLEX10.1 (ILOG) is used to solve LP problems of $LRSP$. An Intel Pentium(R) 4 3.2GHz Processor with 1GB memory is used for computation.

TABLE I
PARAMETERS FOR PROBLEM INSTANCES

| Parameters | |
|---|---|
| number of jobs ($N$) | 50 |
| number of stages ($L$) | 3 |
| due date ($d_i$) | random number on [0, P] |
| weight ($w_i$) | random number on [1, 10] |

### B. Comparison of column generation and LR

Table II shows the comparison of performance between the column generation (CG) and Lagrangian relaxation (LR). UB, LB, DGAP, and Time indicate upper bound, lower bound, duality gap, and computation time, respectively. The quality of lower bound is measured by DGAP = $\frac{UB-LB}{LB} \times 100$. The duality gap for CG is better than that of LR for all cases. On the other hand, the computation time for CG is much larger than that of LR. The average CPU time for CG is approximately 10 times of that of LR for 3 stage flowshop with 50 jobs. This is because the number of columns and simplex multipliers are increased exponentially in the restricted master problem for CG with the increase of the size of problem. It takes much CPU time for solving LP problem repeatedly. From these results it is demonstrated that the quality of lower bound for CG is better than that of LR. However, CPU time for CG is significantly increased with the increase of problem size.

TABLE II
COMPARISION OF COLUMN GENERATION AND LAGRANGIAN RELAXATION ($N = 50$)

| Case | Method | UB | LB | DGAP(%) | Time(s) |
|---|---|---|---|---|---|
| 1 | CG | 1699 | 1533.9 | 10.76 | 160.39 |
| | LR | | 1521.5 | 11.67 | 40.47 |
| 2 | CG | 2843 | 2526.2 | 12.54 | 657.08 |
| | LR | | 2499.2 | 13.76 | 40.59 |
| 3 | CG | 6326 | 5644.3 | 12.08 | 440.00 |
| | LR | | 5641.5 | 12.13 | 40.66 |
| 4 | CG | 7948 | 6791.4 | 17.03 | 353.95 |
| | LR | | 6790.2 | 17.05 | 39.86 |
| 5 | CG | 2545 | 2175.1 | 17.01 | 248.23 |
| | LR | | 2163.9 | 17.61 | 40.59 |
| 6 | CG | 3456 | 2881.7 | 19.93 | 383.42 |
| | LR | | 2878.8 | 20.05 | 42.05 |
| 7 | CG | 3069 | 2749.6 | 11.62 | 176.80 |
| | LR | | 2748.1 | 11.68 | 40.27 |
| 8 | CG | 4787 | 4237.4 | 12.97 | 979.44 |
| | LR | | 4229.7 | 13.18 | 41.09 |
| 9 | CG | 3584 | 3189.9 | 12.35 | 676.56 |
| | LR | | 3187.5 | 12.44 | 40.28 |
| 10 | CG | 1257 | 1137.8 | 10.48 | 152.88 |
| | LR | | 1080.8 | 16.36 | 40.39 |
| Ave | CG | 3751.4 | 3286.7 | 13.68 | 422.87 |
| | LR | | 3274.0 | 14.59 | 40.63 |

### C. Effectiveness of integrated column generation and LR

In this section, the performance of the integrated column generation and LR (CG-LR) is evaluated. The lower bound obtained by CG and CG-LR is all the same. Therefore the performance is evaluated by the number of iterations (CG-Iteration), total computation time (Time) and the number of columns generated (Number of columns). CPU time for CG-LR is the sum of CPU time of LR step (Step 0 of the algorithm in Section IV-D) and CG step (Step 1 to Step 4 of the algorithm in Section IV-D). The parameter $\varepsilon = 0.1$ is used. The effects of parameters $\varepsilon$ to the performance are investigated in Section V-D. The computational results are summarized in Table III.

From the results, the average 25% of total CPU time for CG-LR is successfully reduced in 8 cases of 10 instances except Case 1 and Case 10. Especially, the CPU time reduction is dominant for the cases when the CPU time for conventional CG takes more than 600 seconds with 160 times of iterations. In the column generation, computation time for solving LP is expensive when the number of columns is increased. Therefore the reduction of CPU time is much effective for the instances with a number of iterations required. However, the reduction is not effective for the cases when the conventional CG requires less number of iterations (Case 1 and Case 10). In these two cases, the number of columns is increased more than 1000. Then the CPU time for solving LP is much increased. It is required to adjust parameters for these problems for the proposed method.

TABLE III
PERFORMANCE EVALUATION OF THE PROPOSED METHOD

| Case | Method | Time(s) LR | CG | Total | CG-Iteration | Number of columns |
|---|---|---|---|---|---|---|
| 1 | CG | — | 160.39 | 160.39 | 105 | 3405 |
| | CG-LR | 40.47 | 176.56 | 217.08 | 70 | 4766 |
| 2 | CG | — | 657.08 | 657.08 | 181 | 6211 |
| | CG-LR | 40.59 | 419.06 | 459.67 | 136 | 5461 |
| 3 | CG | — | 440.00 | 440.00 | 138 | 5425 |
| | CG-LR | 40.66 | 287.34 | 328.02 | 87 | 4760 |
| 4 | CG | — | 353.95 | 353.95 | 156 | 5744 |
| | CG-LR | 39.86 | 158.50 | 198.39 | 101 | 3942 |
| 5 | CG | — | 248.23 | 248.23 | 115 | 4554 |
| | CG-LR | 40.59 | 203.49 | 244.09 | 79 | 4890 |
| 6 | CG | — | 383.42 | 383.42 | 163 | 5462 |
| | CG-LR | 42.05 | 235.50 | 277.58 | 130 | 4905 |
| 7 | CG | — | 176.80 | 176.80 | 154 | 4394 |
| | CG-LR | 40.27 | 102.88 | 143.16 | 87 | 3788 |
| 8 | CG | — | 979.44 | 979.44 | 195 | 7306 |
| | CG-LR | 41.09 | 507.36 | 548.45 | 150 | 5952 |
| 9 | CG | — | 676.56 | 676.56 | 165 | 6184 |
| | CG-LR | 40.28 | 426.89 | 467.20 | 118 | 5359 |
| 10 | CG | — | 152.88 | 152.88 | 102 | 3818 |
| | CG-LR | 40.39 | 230.49 | 270.89 | 81 | 4947 |
| Ave | CG | — | 422.87 | 422.87 | 147.4 | 5250.3 |
| | CG-LR | 40.63 | 274.81 | 315.45 | 103.9 | 4877.0 |

### D. Effects of parameter $\varepsilon$ to the performance

In this section, the effects of parameter $\varepsilon$ to the performance of the proposed method is investigated. The performance of CG-LR is evaluated when the parameter $\varepsilon$ is changed as $10^{-4}$, $10^{-3}$, $10^{-2}$, 0.1, 1.0, 10. Table IV shows the effects of parameter $\varepsilon$ to the performance of CG-LR. In Table IV, basic rate indicates the rate of basic columns generated before the execution of column generation. It is given by (number of base columns for the initial columns)/(number of base columns at final condition) $\times$ 100.

From the computational results, the number of iterations is reduced due to the increase of the number of base columns and the number of initial columns when $\varepsilon$ is slightly increased.

However, the total computation time and the number of columns has the minimum when $\varepsilon = 10^{-1}$. The reduction of CPU time is not effective when the base ratio is decreased due to the effects of sufficiently small $\varepsilon$. On the other hand, it takes much CPU time when $\varepsilon$ is large due to the increase of computation time for LP solver. The appropriate value of $\varepsilon$ is different for each case.

TABLE IV
EFFECTS OF PARAMETER $\epsilon$ TO THE PERFORMANCE OF CG-LR

| $\varepsilon$ | Time(s) | CG-Iteration | Iterations Columns | (Initial columns) | Base Rate (%) |
|---|---|---|---|---|---|
| $10^{-4}$ | 447.03 | 138.2 | 5074.0 | (223.1) | 8.82 |
| $10^{-3}$ | 356.16 | 118.2 | 4691.9 | (650.0) | 23.93 |
| $10^{-2}$ | 330.98 | 111.2 | 4890.4 | (1155.9) | 32.04 |
| $10^{-1}$ | 315.45 | 103.9 | 4877.0 | (1378.0) | 36.20 |
| 1 | 318.38 | 95.2 | 4957.9 | (1846.8) | 42.47 |
| 10 | 363.51 | 82.0 | 5811.2 | (3248.9) | 45.15 |

The transition of the value of objective function for Case 4 and Case 10 is shown in Fig. 2, Fig. 3, respectively when the CPU time of Lagrangian relaxation is neglected. In Fig. 2, CPU time for CG-LR is successfully less than half of CPU time for CG when $\varepsilon = 1.0$. However, in Fig. 3, CPU time for LR-CG is worse than that for LR when $\varepsilon = 0.1$. The appropriate determination of the parameter $\varepsilon$ is necessary for CG-LR.

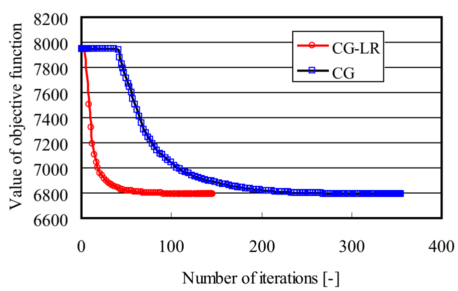


Fig. 2. Transition of value of objective function for Case 4

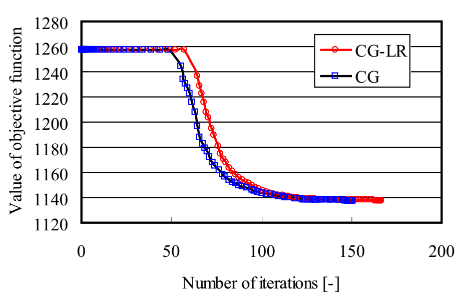


Fig. 3. Transition of the value of objective function for Case 10

## VI. CONCLUSION

In this paper, an integrated column generation and Lagrangian relaxation for flowshop scheduling problems has been proposed. In the proposed method, Lagrangian relaxation is executed before the execution of column generation. The derived near-optimal Lagrange multipliers are used to derive the near-optimal dual solution for continuous relaxation of set partitioning formulation for the original problem. The algorithm of generating base columns has been proposed. The base columns are used as initial columns for column generation. Computational results have demonstrated that the proposed method can reduce approximately 25% of CPU time for flowshop scheduling with 50 jobs and 3 stages compared with the conventional column generation. The appropriate setting of parameter $\varepsilon$ is required for the proposed method. The efficiency of integration of column generation and Lagrangian relaxation depends on the accuracy of Lagrange multiplies derived by Lagrangian relaxation. If the Lagrange multipliers are far from optimal solution, the computation time of column generation becomes much larger than the conventional column generation. In our future work, the estimation of appropriate parameter $\varepsilon$ will be studied. The applicability of the proposed methodology to large-sized problem will be investigated.

## REFERENCES


[1] M. Nawaz, E.E. Enscore and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *OMEGA*, vol. 11, pp. 91–95, 1983.

[2] C. Chung, J. Flynn, K. Ömer, "A branch and bound algorithm to minimize the total tardiness for $m$-machine permutation flowshop problems," *European Journal of Operational Research*, vol. 174, no. 1, pp. 1–10, 2006.

[3] O. Moursli, Y. Pochet, "A Branch-and-bound algorithm for the hybrid flowshop," *International Journal of Production Economics*, vol. 64, pp. 113–125, 2000.

[4] V. Armentano, D. Ronconi, "Tabu search for total tardiness minimization in flowshop scheduling problems," *Computers and Operations Research*, vol. 26, no. 3, pp. 219–235, 1999.

[5] R. Riuz and T. Stützle, "An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives," *European Journal of Operational Research*, vol. 187, no. 3, pp. 1143–1159, 2008.

[6] D. Huisman, "A column generation approach to the rail crew re-scheduling problem," *European Journal of Operational Research*, vol. 180, no. 1, pp. 163–173, 2007

[7] E. Choi, D.W. Tcha, "A column generation approach to the heterogeneous fleet vehicle routing problem," *Computers and Operations Research*, vol. 34, no. 7, pp. 2080–2095, 2007.

[8] M. Lamiri, V. Augusto and X. Xie, "Patients scheduling in a hospital operating theatre," *Proceedings of the 4th IEEE Conference on Automation Science and Engineering*, pp. 627–632, 2008.

[9] M.E. Lübbecke and J. Destosiers, "Selected topics in column generation," *Operations Research*, vol. 53, no. 6, pp. 1007–1023, 2005.

[10] J.M. van den Akker, J.A. Hoogeveen and S.L. van de Velde, "Combining column generation and Lagrangian relaxation to solve a single-machine common due date problem," *INFORMS Journal on Computing*, vol. 14, no. 1, pp. 37–51, 2002.

[11] D. Huisman, R. Jans, M. Peeters and A.P.M. Wagelmans, "Combining column generation and Lagrangian relaxation," *ERIM Report Series*, Reference No. ERS-2003-092-LIS, 2003.

[12] G.L. Nemhauser and L.A. Wolsey, "Integer and combinatorial optimization," *Wiley-Interscience*, pp. 323–336, 1988.

[13] T. Nishi, Y. Hiranaka and M. Inuiguchi, "Successive Lagrangian approach for flowshop scheduling problems to minimize the total weighted tardiness," *Proceedings of the 3rd Annual Conference on Automation Science and Engineering*, pp. 658–663, 2007.

[14] E. Taillard, "Benchmarks for basic scheduling problems," *European Journal of Operational Research*, vol. 64, no. 2, pp. 278–285, 1993.