Parallel Machine Scheduling under the Disruption of Machine Breakdown

Lixin Tang* and Yanyan Zhang

Liaoning Key Laboratory of Manufacturing System and Logistics, The Logistics Institute, Northeastern University, Shenyang, 110004, China

This paper investigates the scheduling problem of dealing with disruption caused by machine breakdown under the environment of identical parallel machines. When a machine breakdown changes the scheduling environment, the original schedule cannot be executed according to the prescribed plan. The objective after disruption is to create a recovery schedule that minimizes the original objective function and the deviation from the original schedule. The original objective function is the total weighted completion time, and the deviation is the completion time of jobs from the planned one. The recovery scheduling problem is formulated as an integer programming model. A Lagrangian relaxation (LR) framework that relaxes the machine capacity constraints is proposed for solving the model. The subgradient method is used to update the values of Lagrangian multipliers. The extensive computational experiments are carried out, and the numerical results demonstrate that the proposed LR approach based on the developed model can provide an excellent recovery schedule solution in a timely manner.

1. Introduction

Parallel machine scheduling problems have been intensively studied for decades. However, most efforts have been devoted to regular research that assumes that the scheduling environment is static and deterministic. While in practice, production schedules are usually performed under disruptions with high levels of uncertainty such as machine breakdown, supply shortage, worker unavailability, unplanned demand changes, market competence, and the commitment of upper management. The original schedule, even though it is optimal with respect to the primal environment, will be no longer suitable for the new environment. These disruptions inevitably result in deviations from the original schedule. Disruption management is associated with how to deal with some unforeseen events which occurr during the execution process of a schedule. The literature shows that disruption management has been successfully applied in many areas, such as airline scheduling, production planning, and scheduling in process industries.

Among the applied areas of disruption management, the airline industry has been one of the most active ones.[1−4] The performance of an airline largely depends on how well it can follow its published schedule. Constant disruptions due to mechanical failures, personnel shortages, and severe weather can play havoc with the equipment, throwing flight schedules off for hours or days.[5] Love et al.[1] investigated the aircraft recovery problem when unforeseen events have disrupted the existing flight schedule and developed a heuristic approach to generate feasible revised flight schedules of a good quality. Yu et al.[2] developed a decision-support system for airlines to generate globally optimal, or near optimal, crew-recovery solutions when published schedules are delayed or canceled by unexpected events such as inclement weather, aircraft mechanical problems, and crew unavailability. Argüello et al.[3] presented a greedy randomized adaptive search procedure to reconstruct aircraft routings in response to groundings and delays experienced over the course of the day. Thengvall et al.[4] considered the recovery scheduling of all aircraft operated by a large carrier following a hub closure. Three multicommodity network-type models for determining a recovery schedule are presented, each allows for cancellations, delays, ferry flights, and substitution between fleets and subfleets.

The disruption management in other areas such as chemical, iron, and steel industries is also necessary and important. In the chemical industry, some previous work has been reported on the rescheduling framework considering various disturbances. Vin and Ierapetritou[6] developed a two-stage solution approach for the problem of reactive scheduling in multiproduct batch plants. Two kinds of disturbances are considered: machine breakdown and rush order arrival. The optimal new schedule is obtained from the solution of a mixed integer linear programming (MILP) formulation that systematically incorporates all different rescheduling alternatives. Roslöf et al.[7] introduced a MILP based algorithm to improve an existing nonoptimal production schedule or to reschedule jobs in the case of changed operational parameters. Méndez and Cerdá[8] presented an MILP formulation for reactive scheduling of multiproduct batch plants to modify schedules when unforeseen events occur. These events include batch reordering, unit reallocation, and insertion of new batches. In their approach, multiple rescheduling operations can be performed at the same time. Later, they[9] extended this model to resource-constrained multistage batch facilities, in which the availability of renewable manpower resources is taken into account. Méndez et al.[10] reviewed the current reactive capabilities for short-term batch process scheduling and analyzed the future research in this field.

As for the situation in the iron and steel industry, there is usually more than one machine at the stage of iron-making and continuous casting. It is well-known that the most special feature of iron and steel production is the temperature requirement at each operation. Temperature decrease caused by production delay will result in huge undesirable cost. Once a schedule is established, the assignment and order of each charge (the scheduling unit) are determined, and the link between operations along the production line is formed. Completing earlier or later than the prescribed time will destroy production continuity and lead to unnecessary cost. In practice, production disruption occurs frequently and unavoidably, and then, generating a new schedule with the best performance and lowest deviation has been a challenging task. The goal of each stage is to complete

* To whom correspondence should be addressed. Tel.: +86-24-83680169. Fax: +86-24-83680169. E-mail: qhjytlx@mail.neu.edu.cn (L.T.).

10.1021/ie801868f CCC: $40.75
Published on Web 06/17/2009

the production task at the planned time. However, according to our investigation, most of these situations are tackled by personal experience, by which the quality of the new schedule cannot be guaranteed.

Generally, production planning and scheduling problems can be reduced to machine scheduling problems and some results on the disruption management of machine scheduling have been reported. Gawiejnowicz[11] considered two problems of scheduling a set of independent, nonpreemptable, and proportionally deteriorating jobs on a single machine with constraints on the availability of the machine or on jobs and the maximum completion time criterion. They proved that these problems are nonpolynomial (NP) complete in the ordinary sense or in the strong sense, depending on the number of nonavailability periods or the number of distinct ready times and deadlines. Kasap et al.[12] investigated optimal sequencing policies for the expected makespan problem where jobs have to be reprocessed in their entirety if preemptions occur because of breakdowns. Liao and Sheen[13] proposed a binary search algorithm to schedule n independent jobs on m identical machines incorporating machine availability and eligibility constraints to minimize the makespan. The algorithm can either verify the infeasibility of the problem or solve it optimally in a polynomial time if a feasible schedule exists. Liao and Chen[14] developed a heuristic approach for a problem encountered in a textile company where the machine breakdown occurs frequently. The basic idea of their approach is to provide a longer setup time to reduce the breakdown rate. The effectiveness of the heuristic is evaluated by comparing its solution with both the solutions of the current scheduling method and the branch-and-bound algorithm using real data from the company. Lee and Yu[15] designed optimal algorithms respectively for minimizing the expected total weighted completion times and maximizing the expected tardiness on single machine, where a disruption will happen with a possibility at a particular time and last for a period of time with a certain probability. Lee et al.[16] took two machines as an example to study a scheduling problem when some machines become unavailable for certain periods. The objective is to reschedule jobs to minimize the original cost function and possibly transportation costs and disruption cost caused by deviating from the originally planned completion times. For each problem, they can either provide a polynomial algorithm for solving or show its NP-hardness and present a pseudopolynomial algorithm for the NP-hard problem in the ordinary sense. Qi et al.[17] addressed the problem of updating a machine schedule when a disruption occurs after a subset of the jobs has been processed. Both single machine and parallel two-machine problems were analyzed using a variety of deviation cost measures and rescheduling policies.

In this paper, we study the problem of how to react when a machine disruption occurs. A machine disruption is said to occur when a machine becomes unavailable for some period of time, no matter which jobs are scheduled during the period. Such disruption, if not managed properly and timely, will severely affect the schedule performance in terms of operational efficiency and lead to inconvenience. At the time of a disruption, certain options may be available: Jobs that are assigned to the disrupted machines and have not yet been completed can either be processed by the same machine after the disruption or can be moved to other available machines for processing. In practice, once a schedule is established, many preparations such as material handling are laid out. The larger the deviation is, the more effort will be spent to change the preparations for the new schedule. Then, the former options can only be adopted when the length of breakdown time is small enough compared with processing time so that it can be ignored, the initial schedule may still be followed with little deviation. In most cases, the deviation induced by the disruption is so serious that the initial schedule is no longer optimal or even infeasible with respect to the original problem; therefore, a new schedule is needed.

As an emerging research area, most of the reported results on disruption management either focus on the problem that can be originally solved to optimality[17] or the recovery schedule is obtained by heuristics.[18,19] To the best of our knowledge, little research has addressed the optimization-based algorithms for disruption management of the originally NP-hard problem. The primary purpose of this paper is to provide an optimization-based method to deal with this type of problem. A major contribution of the work is the development of a Lagrangian relaxation (LR) algorithm based recovery method, which tries to be suitable for the new environment, in the sense that the schedule is optimal or near with respect to the original objective function and the deviation penalties so that the new schedule is not too far away from the original schedule.

In the next section, the problem statement and disruption description are presented, and then an integer linear programming model is developed for the recovery scheduling problem. In the section thereafter, we present the solving procedures of Lagrangian relaxation algorithm for the problem. Computational experiments are given in the next section followed by the summary and analysis of the computational results. The conclusion is presented in the last section.

2. Problem Description and Formulation

2.1. Problem Statement and Definitions. The goal of parallel machine scheduling is to develop a detailed schedule specifying assignment and start time subject to resources constraints. In the case of machine breakdown, besides the above constraints, we seek a new recovery schedule that will be close to the original schedule so as not to cause too much inconvenience under the changed environment.

The disruption management problem under investigation can be described as follows. Originally, a set of jobs is to be scheduled on a number of identical parallel machines so that a given objective function, the sum of weighted completion times, is minimized. Each job can be processed on any of the identical machines. While the schedule on the original problem (original schedule) is being executed, a machine breakdown occurs so that the original schedule is no longer feasible because the jobs executed not yet on this machine need reassignment. We assume that at most one disruption takes place at a time. It is also assumed that the disruption lasts only finitely long, and the breakdown time and recovery time are known. Rescheduling is performed after all ongoing jobs at the disruption time are all completed. All jobs that start later than the breakdown time need be rescheduled. It is assumed that there are n unexecuted jobs, we will reindex them from 1 to n. After the disruption occurs, the earliest start time at which all the ongoing jobs have completed and the rescheduling can be executed is set to be 1. In the original schedule each job j has a ready time r_j and a planned completion time $\bar{C}_j$, whose values are revised according to the new time index. In addition, it is assumed an optimal schedule has been achieved before the disruption.

How well the new (recovery) schedule suits the new environment should not be the sole criterion to measure the schedule's effectiveness, since there is also a considerable amount of penalty associated with the new schedule's deviation from the initial one. If a planned schedule is modified too much, just as

pointed by Brown et al.,[20] a common complaint of managers using optimization-based decision support systems is that small changes in input often translate into drastically different solutions. Therefore, the undesirable deviation should be as important as the optimization itself. Even though it is impossible to make the recovery schedule to completely converge to the initial one, it will be more preferable if the deviation of the new schedule can be decreased as small as possible. Therefore, the objective function adopted in this paper considers the total weighted completion time of the new schedule that is similar to the original objective function and the penalty incurred for both positive and negative deviations from the completion time of the original schedule.

Let $\bar{C}_j$ be the completion time of job j in the original schedule. If no disruption occurs, job j will finish processing at this time so in the rescheduling problem we can view $\bar{C}_j$ as a special type of due date. If C_j is the real completion time in the new schedule, then the difference between $\bar{C}_j$ and C_j can be used to assess the deviation cost. A common approach[14,16−18,21] has been used to formulate the recovery problem, in which an virtual earliness and virtual tardiness penalty are prescribed based on the difference of job completion times $\bar{C}_j$ and C_j. The definition of the virtual earliness is $E_j = \max\{0, \bar{C}_j - C_j\}$ and the virtual tardiness $T_j = \max\{0, C_j - \bar{C}_j\}$. If a job is completed earlier than planned, additional inventory holding cost proportional to earliness may occur; similarly, if a job is completed later than planned, the manufacture may have to pay for lateness cost. Therefore, completing earlier or later than planned time is discouraging.

2.2. Formulation of Recovery Problem. When modeling, the time horizon is partitioned into a set of time intervals of unit length, that is, the formulation is established based on a discrete approach. See Notations for a description of the terms used below.

The integer linear program with the composite objective of minimizing total weighted completion time and deviations from the original schedule is as follows.

Objective function

$$\underset{\{C_j\}}{\text{minimize}} \sum_{j=1}^{n} w_j C_j + \sum_{j=1}^{n} \alpha(C_j - \bar{C}_j)^+ + \sum_{j=1}^{n} \beta(\bar{C}_j - C_j)^+ \tag{1}$$

subject to

$$\sum_{j=1}^{n} x_{jt} \le m \qquad t = 1, 2, ..., T \tag{2}$$

$$\sum_{t=1}^{T} x_{jt} = p_j \qquad j = 1, 2, ..., n \tag{3}$$

$$t x_{jt} \le C_j \quad j = 1, 2, ..., n;\ t = 1, 2, ..., T \tag{4}$$

$$C_j - p_j + 1 \le t + M(1 - x_{jt}) \quad j = 1, 2, ..., n;\ t = 1, 2, ..., T \tag{5}$$

$$C_j - p_j + 1 \ge r_j \quad j = 2, 3, ..., n \tag{6}$$

$$x_{jt} \in \{0, 1\} \quad j = 1, 2, ..., n;\ t = 1, 2, ..., T \tag{7}$$

$$C_j \in \{1, 2, ..., T\} \quad j = 1, 2, ..., n \tag{8}$$

Although there are some similarities between the scheduling problem that needs be completed after a disruption and the original one, the differences are significant because different performance requirements are considered during the recovery process. In the above formulation, a completely different schedule emerges with the new objective function expressed by eq 1, the total weighted completion time of all jobs plus the penalty cost for deviation from the initially planned completion time. Constraints 2 ensure that at any time point t there are at most m jobs that can be processed, which relates to the machine capacity constraints. Constraints 3 ensure the time intervals over which a job requires a machine. Constraints 4 guarantee that a job occupies one of the machines until the processing is finished. Constraints 5 present the requirement of the start time of a job. Constraints 6 require that a job can only begin processing after it is ready. Finally, constraints 7 and 8 define the range of the variables.

In the above formulation, m is the number of current available machines. This formulation can be applied to different situations of disruption. That is to say, how long the disruption will last, whether or not the disrupted machine can recover within the horizon, and how many machines become unavailable simultaneously, the only difference in our formulation is the value of m.

3. Solution Methodology: Lagrangian Relaxation Framework

3.1. Overview. Lagrangian relaxation has been successfully applied to many complex scheduling problems.[22,23] It is a practical approach based on the idea of decomposition and coordination to obtain near optimal schedules for the hard-to-solve problems. In the LR, after coupling constraints are relaxed by the lagrangian multipliers, the relaxed problem can be decomposed into a set of easier subproblems. The subproblems are not NP-hard and can be easily solved. The coordination of subproblem solutions is performed through iterative updating to reduce coupling constraint violations. At the end of such iterations, a simple heuristic is constructed to convert subproblem solutions into a feasible schedule satisfying all constraints. In this paper, when machine capacity constraints 2 are relaxed, the formulation can be expressed as separable sets of jobs, and the overall problem is decomposed into job-level subproblems which can be solved by using an enumerative algorithm. Coordination of subproblem solutions is achieved through the iterative updating of multipliers based on the subgradient concept.

3.2. Forming the Relaxed Problem. Lagrangian multipliers $\{u_t\}$ are introduced to relax the coupling constraints 2 and incorporate it into the objective function. The relaxed problem is defined as follows.

$$\underset{\{C_j\}}{\text{minimize}}\ \text{LR, with}$$

$$\text{LR} = \sum_{j=1}^{n} w_j C_j + \sum_{j=1}^{n} \alpha(C_j - \bar{C}_j)^+ + \sum_{j=1}^{n} \beta(\bar{C}_j - C_j)^+ + \sum_{t=1}^{T} u_t \left(\sum_{j=1}^{n} x_{jt} - m\right) \tag{9}$$

The relaxed problem is to minimize the objective function 9 subject to operation processing constraints 3−5, ready time constraints 6, variable range constraints 7 and 8, and the following multipliers constraints.

$$u_t \ge 0, \quad t = 1, 2, ..., T \tag{10}$$

3.3. Formulation and Resolution of Subproblems. After coupling constraints 2 are relaxed, all terms including the objective function

and the constraints can be represented as individual sets of jobs. The subproblem for one job is thus formulated as eq 11.

$$\begin{aligned} &\text{minimize } \mathrm{LR}_j, \text{ with} \\ &\mathrm{LR}_j = w_j C_j + \alpha(C_j - \bar{C}_j)^+ + \beta(\bar{C}_j - C_j)^+ + \sum_{t=1}^{T} u_t(x_{jt} - m) = w_j C_j + \alpha(C_j - \bar{C}_j)^+ + \\ &\beta(\bar{C}_j - C_j)^+ + \sum_{t=1}^{T} u_t x_{jt} - \sum_{t=1}^{T} u_t m \end{aligned} \tag{11}$$

The minimization is subject to constraints 3−8 and 10. The subproblems are job-level, that is, each subproblem includes just one job. The problem variables in each subproblems are C_j and x_{jt}, that is, to decide the start time/completion time of each job with provided values of multipliers from the previous iteration. The subproblem can be solved by using an enumerative algorithm; each time point from ready time to the end of the time horizon is enumerated as the possible start time, among which the time that leads to the best relaxed objective value is selected as the real start time. The multipliers $\{u_t\}$ here play a key role of coordination by the updated value at each iteration, which is adjusted by subgradient method according to the degree of constraints violation in the previous iteration.

3.4. Solving the Dual Problem. The goal of the dual problem is to find an optimal set of multipliers that maximizes the following dual function.

maximize LD, with

$$\begin{aligned} &\underset{u_t}{\text{maximize LD, with}} \\ &\mathrm{LD} = \sum_{j=1}^{n} w_j C_j + \sum_{j=1}^{n} \alpha(C_j - \bar{C}_j)^+ + \sum_{j=1}^{n} \beta(\bar{C}_j - C_j)^+ + \sum_{t=1}^{T} u_t\Big(\sum_{j=1}^{n} x_{jt} - m\Big) \end{aligned} \tag{12}$$

subject to eq 10.

The optimal start times obtained from the subproblems solutions are used when solving the dual problem 12. The subgradient optimization method is used in this paper. The subgratient direction of Lagrangian multipliers $\{u_t\}$ is updated as follows.

$$u^{h+1} = u^h + \alpha^h g(u^h) \tag{13}$$

Where h is the iteration index, and $g(u)$ is the subgradient of L at the hth iteration. The subgradient component at time t is equal to $(\sum_{j=1}^{n} x_{jt} - m)$. The step size α^h at iteration h is set to be

$$\alpha^h = \lambda \frac{L^* - L^h}{||g(u^h)||^2}, \quad 0 < \lambda < 2 \tag{14}$$

Where L^* is the objective value of optimal feasible solution and L^h is the objective value of the dual function at the hth iteration. Since L^* is unknown, it is usually estimated by the best feasible objective value found so far. Parameter λ used here is initialized with 0.5. It will be adjusted if L^h remains approximately the same value for a fixed number of iterations.

3.5. Designing Heuristics to Obtain Feasible Solutions. Subproblem solutions, when put together, are generally infeasible since coupling constraints 2 have been relaxed. A heuristics is, therefore, developed to convert such solutions into feasible ones. The procedure is illustrated as follows.

Table 1. Data Used in the Example

job	1	2	3	4	5	6	7	8	9	10
r	31	0	30	56	104	54	140	83	149	117
p	22	21	44	24	40	27	47	43	39	47
w	4	9	2	1	8	8	6	8	8	1
$\bar{C}$	52	21	73	97	143	80	186	125	187	233

First, a list is created in the ascending order of the assigned optimal start times obtained from subproblems.

According to the list, jobs are assigned to the available machine at that time point.

When multiple jobs competed for a less number of machines at a particular time unit and have the same processing time, weights of these jobs are sequenced. Jobs are then assigned to machines in the descending order of their weights.

If more than m (the number of available machines) jobs are being processed at some interval t, those jobs with the lowest weights (priorities) are postponed.

The process terminates if all jobs are scheduled.

4. Computational Experiments

4.1. Data Set Generated. The proposed method is implemented on a PC with Pentium-IV (2.26 GHz) CPU with 1024 SDRAM using the language C++ and the Vista operating system. The number of jobs to be rescheduled is from 20 to 160. The total number of machines (available and unavailable) is from 2 to 10. For each job, an integer processing time is randomly generated between 20 and 50 and an integer weight is uniformly distributed in the interval (1, 10). Ready times are generated randomly using the approach proposed by Gélinas and Soumis.[24] A schedule is constructed by selecting jobs at random and then scheduling them at the earliest possible time t_i. This schedule is used to generate an integer ready time $r_i = t_i - U[1, 50]$. If the ready time of any job is negative, such adjustment is made: $r_i = \max\{r_i, 0\}$. The combination of the number of jobs and machines forms 225 problem scenarios; 10 different problem instances of the same scenario are randomly generated. Thus, in total, 2250 problem instances are tested in the experiment.

It is assumed that there is only one machine that becomes unavailable. The length of breakdown is randomly generated in the interval of 1 and the end of time horizon. Parameters α and β, the penalties of virtual tardiness and earliness of each job, are set to be 0.5, 0.3, and 0.7, respectively. For LR-based methods, the subgradient method is used to update the multipliers with the initial value of 0. The updating process of multipliers stops either the duality gap is smaller than the predefined value (here is 0.5%) or the number of iteration reaches the maximum value (here is 400). The duality gap, the distance between the upper bound and the lower bound of a problem, is one of the performance measures besides the computation time to evaluate the proposed LR algorithm. The lower bound is obtained by the computing the objective value of the dual function, and the upper bound is the objective value of a feasible schedule. The computation of the duality gap is as follows.

$$\text{gap} = \frac{\mathrm{UB} - \mathrm{LB}}{\mathrm{LB}} \times 100\%$$

where UB is the best upper bound ever found when the algorithm ends, and LB is the best lower bound.

4.2. Computational Results and Analysis. First, we give an example to show the efficiency of the proposed approach; we generated an instance including 10 jobs and 2 machines and tested

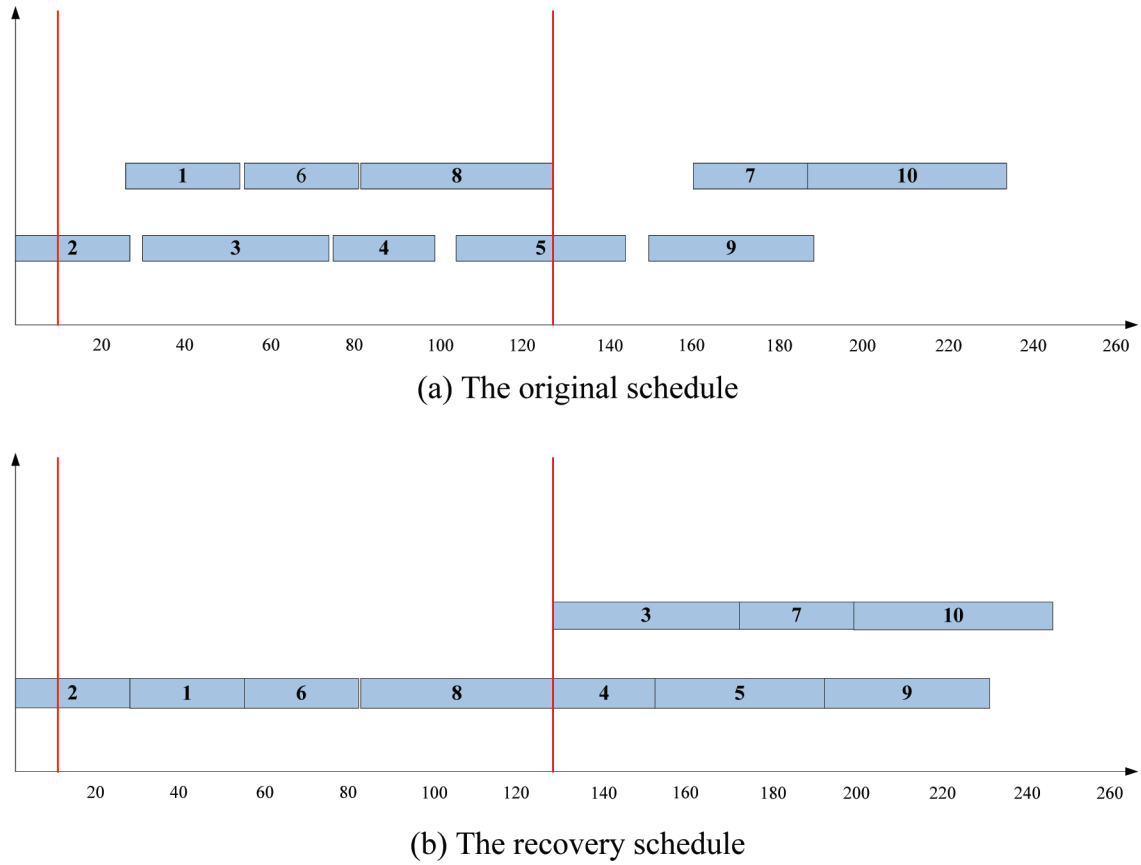

Figure 1. Gantt chat illustration of the original schedule and the recovery schedule.

Table 2. Summary of Duality Gap When the Disrupted Machine Can Be Made Available after Some Period of Time

M	$\alpha = 0.3\ \beta = 0.7$					$\alpha = 0.5\ \beta = 0.5$					$\alpha = 0.7\ \beta = 0.3$				
N	2	4	6	8	10	2	4	6	8	10	2	4	6	8	10
20	0.29	0.13	0.00	0.00	0.00	0.31	0.13	0.00	0.00	0.00	0.32	0.14	0.00	0.00	0.00
30	0.44	0.20	0.05	0.05	0.00	0.44	0.21	0.05	0.05	0.00	0.44	0.21	0.06	0.05	0.00
40	0.37	0.22	0.11	0.00	0.00	0.38	0.21	0.11	0.00	0.00	0.39	0.25	0.12	0.00	0.00
50	0.46	0.25	0.12	0.07	0.06	0.44	0.25	0.13	0.10	0.06	0.45	0.26	0.15	0.07	0.06
60	0.30	0.24	0.25	0.06	0.05	0.35	0.23	0.27	0.07	0.05	0.32	0.24	0.28	0.08	0.06
70	0.47	0.26	0.15	0.12	0.08	0.49	0.26	0.15	0.12	0.12	0.48	0.29	0.16	0.13	0.12
80	0.40	0.33	0.25	0.15	0.13	0.42	0.36	0.24	0.16	0.13	0.43	0.36	0.24	0.17	0.14
90	0.46	0.31	0.33	0.17	0.09	0.46	0.31	0.33	0.19	0.09	0.46	0.35	0.35	0.18	0.09
100	0.43	0.36	0.32	0.16	0.16	0.43	0.37	0.32	0.16	0.17	0.44	0.39	0.33	0.17	0.16
110	0.38	0.34	0.22	0.22	0.10	0.39	0.36	0.22	0.24	0.09	0.42	0.37	0.23	0.25	0.11
120	0.40	0.35	0.27	0.22	0.19	0.40	0.37	0.25	0.22	0.20	0.43	0.38	0.27	0.23	0.20
130	0.36	0.36	0.25	0.20	0.17	0.41	0.37	0.25	0.21	0.17	0.42	0.38	0.26	0.24	0.20
140	0.34	0.31	0.25	0.23	0.22	0.36	0.32	0.25	0.23	0.21	0.36	0.35	0.25	0.25	0.22
150	0.33	0.36	0.32	0.27	0.26	0.35	0.37	0.32	0.29	0.27	0.37	0.37	0.32	0.30	0.28
160	0.36	0.31	0.21	0.25	0.19	0.37	0.33	0.22	0.26	0.19	0.38	0.34	0.22	0.27	0.20
avg	0.39	0.29	0.21	0.14	0.11	0.40	0.30	0.21	0.15	0.12	0.41	0.31	0.22	0.16	0.12

it. The adopted data are listed in Table 1; the breakdown time is 10, and the recovery time is 125 (shown by the red lines). The Gantt chart illustration is shown by Figure 1. The gap between jobs in the original schedule is formed by the ready times. The results obtained by LR algorithm have the duality gap of 0.19%.

Then, two experiments are designed to demonstrate the performance of our approach. In experiment 1, it is assumed that the disrupted machine can be available after some period of time. While in experiment 2, it is assumed that the disrupted machine will never be available within the time horizon, implying that the resource constraint becomes harder than before. In both experiments, different values of α and β, the penalties of virtual tardiness and earliness, are tested. The results of total 225 problem scenarios are reported in Tables 2−5 including the duality gap and the running time, in which each data (except the avg) is the average result of 10 instances belonging to the same scenarios.

The data in Tables 2 and 3 show the results of experiment 1, the duality gap and the running time when the disrupted machine can be made available after some period of time. When the recovery time is randomly generated between zero and the end of time horizon, the LR algorithm can obtain solutions with the average duality gap being less than 0.5% for various numbers of machines. For most test problems, our algorithm can find near-optimal solutions with the average running time being about 30 s and the longest one being about 75 s, which coincides with the practical requirement of managing the disruption effectively and quickly.

Table 3. Summary of Running Time When the Disrupted Machine Can Be Made Available after Some Period of Time (unit: seconds)

M	$\alpha = 0.3\ \beta = 0.7$					$\alpha = 0.5\ \beta = 0.5$					$\alpha = 0.7\ \beta = 0.3$				
N	2	4	6	8	10	2	4	6	8	10	2	4	6	8	10
20	1.10	0.27	0.15	0.09	0.05	1.09	0.28	0.15	0.10	0.05	1.17	0.29	0.14	0.09	0.05
30	2.78	0.87	0.42	0.22	0.14	2.79	0.91	0.40	0.25	0.15	2.80	0.91	0.41	0.24	0.15
40	5.11	1.80	0.85	0.44	0.24	5.22	1.88	0.96	0.46	0.25	5.25	1.88	1.08	0.53	0.24
50	7.50	2.72	1.59	1.00	0.46	7.55	2.74	1.65	1.12	0.48	7.67	2.75	1.55	1.09	0.48
60	10.84	3.98	2.38	1.34	0.71	10.80	3.95	2.43	1.21	0.86	10.81	3.97	2.28	1.39	0.90
70	14.29	5.47	3.51	2.25	1.59	14.38	5.46	3.49	2.37	1.37	14.38	5.44	3.48	2.31	1.44
80	18.84	7.07	4.54	2.46	1.91	18.67	7.00	4.53	2.80	2.03	18.82	7.07	4.52	2.55	2.15
90	24.15	9.25	5.66	4.37	2.59	23.98	9.16	5.64	4.07	2.56	24.17	9.17	5.70	4.34	2.64
100	29.41	11.27	7.05	5.16	3.41	29.19	11.13	6.97	4.88	3.46	29.31	11.18	7.00	4.83	3.77
110	36.20	31.37	30.22	29.25	25.83	34.92	31.39	30.12	28.95	26.20	35.13	31.77	30.25	29.18	26.66
120	42.11	37.42	35.26	33.34	30.81	41.76	37.49	34.96	34.28	32.13	42.01	37.65	35.27	34.14	32.25
130	48.65	43.61	41.91	41.25	39.76	48.39	43.23	41.49	41.06	38.91	48.93	43.68	41.52	41.36	39.46
140	57.36	50.60	49.01	47.80	45.12	57.37	50.07	48.13	46.75	46.29	57.31	50.78	48.82	47.46	46.13
150	65.36	56.65	56.96	54.27	52.69	64.94	56.04	56.41	53.50	52.88	65.92	56.96	57.32	54.25	53.71
160	75.16	65.15	62.24	62.53	61.37	74.20	65.07	61.67	61.86	60.71	75.21	65.37	62.09	62.24	59.61
avg	29.26	21.83	20.12	19.05	17.78	29.02	21.72	19.93	18.91	17.89	29.26	21.92	20.10	19.07	17.98

Table 4. Summary of Duality Gap When the Disrupted Machine Cannot Be Made Available until the End of the Time Horizon

M	$\alpha = 0.3\ \beta = 0.7$					$\alpha = 0.5\ \beta = 0.5$					$\alpha = 0.7\ \beta = 0.3$				
N	2	4	6	8	10	2	4	6	8	10	2	4	6	8	10
20	1.02	0.50	0.37	0.44	0.34	1.02	0.50	0.39	0.45	0.32	1.06	0.54	0.39	0.46	0.32
30	1.21	0.63	0.44	0.45	0.47	1.17	0.72	0.43	0.41	0.48	1.13	0.68	0.47	0.43	0.46
40	1.13	1.00	0.44	0.47	0.45	1.12	1.02	0.47	0.47	0.44	1.13	1.17	0.56	0.46	0.43
50	1.49	1.01	0.63	0.55	0.46	1.44	0.99	0.67	0.56	0.48	1.45	1.03	0.65	0.54	0.43
60	1.32	1.09	0.66	0.51	0.50	1.31	1.14	0.67	0.48	0.52	1.28	1.16	0.68	0.55	0.51
70	1.23	1.16	1.06	0.63	0.51	1.19	1.22	1.07	0.68	0.51	1.28	1.20	1.02	0.72	0.51
80	1.26	1.05	0.99	0.50	0.51	1.37	1.04	1.04	0.51	0.55	1.35	1.17	1.06	0.56	0.56
90	1.50	1.17	1.07	0.75	0.53	1.44	1.28	1.17	0.70	0.53	1.38	1.19	1.21	0.72	0.63
100	1.31	1.30	1.08	0.68	0.49	1.22	1.24	1.10	0.71	0.56	1.33	1.41	1.21	0.68	0.56
110	1.39	1.82	1.27	0.76	0.65	1.45	1.87	1.28	0.77	0.68	1.47	1.83	1.42	0.86	0.69
120	1.55	2.20	1.35	0.78	0.53	1.55	2.11	1.22	0.89	0.59	1.62	2.14	1.26	0.81	0.65
130	1.35	2.21	1.51	0.75	0.66	1.46	2.31	1.53	0.78	0.69	1.40	2.37	1.52	0.80	0.72
140	1.54	2.22	1.34	0.86	0.74	1.62	2.33	1.29	0.87	0.76	1.70	2.26	1.30	0.91	0.77
150	1.70	2.15	1.56	0.97	0.61	1.75	2.14	1.57	1.01	0.66	1.77	2.10	1.59	1.04	0.72
160	1.70	2.67	1.91	0.99	0.74	1.65	2.41	1.76	0.98	0.74	1.75	2.33	1.71	1.04	0.83
avg	1.38	1.48	1.05	0.67	0.55	1.38	1.49	1.04	0.68	0.57	1.41	1.51	1.07	0.71	0.59

Table 5. Summary of Running Time When the Disrupted Machine Cannot Be Made Available until the End of the Time Horizon (unit: seconds)

M	$\alpha = 0.3\ \beta = 0.7$					$\alpha = 0.5\ \beta = 0.5$					$\alpha = 0.7\ \beta = 0.3$				
N	2	4	6	8	10	2	4	6	8	10	2	4	6	8	10
20	1.11	0.29	0.16	0.10	0.05	1.09	0.30	0.16	0.10	0.05	1.16	0.30	0.15	0.09	0.05
30	2.77	0.90	0.43	0.23	0.15	2.77	0.95	0.42	0.26	0.16	2.74	0.95	0.42	0.25	0.15
40	5.16	1.84	0.88	0.46	0.25	5.26	1.95	1.00	0.48	0.26	5.26	1.93	1.12	0.56	0.25
50	7.57	2.83	1.68	1.04	0.48	7.59	2.84	1.71	1.17	0.49	7.64	2.89	1.63	1.15	0.50
60	11.19	4.11	2.46	1.40	0.74	11.09	4.09	2.51	1.27	0.90	11.26	4.16	2.38	1.46	0.96
70	14.60	5.69	3.67	2.35	1.64	14.55	5.65	3.64	2.47	1.41	14.66	5.66	3.70	2.42	1.49
80	19.56	7.29	4.74	2.57	2.00	19.47	7.27	4.70	2.93	2.14	19.61	7.33	4.73	2.68	2.26
90	24.79	9.60	5.89	4.54	2.74	24.42	9.43	5.81	4.21	2.66	24.67	9.58	5.89	4.52	2.77
100	30.38	11.63	7.28	5.37	3.55	30.21	11.50	7.20	5.07	3.60	30.32	11.62	7.27	5.04	3.91
110	36.44	33.08	31.61	30.69	27.19	36.37	32.82	31.48	30.28	27.62	36.49	33.05	31.58	30.71	28.10
120	43.70	39.31	37.31	35.10	32.44	43.36	39.10	36.88	36.21	34.18	43.64	39.26	37.28	35.99	33.78
130	50.87	45.44	43.84	43.37	41.84	50.99	45.08	43.43	43.33	40.88	50.48	45.13	43.55	43.53	41.54
140	60.09	52.53	50.88	49.57	47.47	59.76	51.99	51.30	49.07	48.27	59.80	52.71	51.04	49.65	47.78
150	68.46	59.08	58.88	56.07	54.99	67.58	58.85	58.87	56.13	55.19	68.31	59.54	59.25	56.05	56.08
160	77.49	68.65	64.79	64.82	64.02	77.04	67.98	64.18	64.06	63.58	77.88	68.56	65.24	64.53	62.81
avg	30.28	22.82	20.97	19.85	18.64	30.10	22.65	20.89	19.80	18.76	30.26	22.84	21.02	19.91	18.83

The data in Tables 4 and 5 show the results of experiment 2, the duality gap and the running time when the disrupted machine will never be made available by the end of the time horizon. The generated recovery schedule is acceptable with the largest average duality gap being 1.51%.

For the situation in experiment 1, we also investigate the difference in objective function between the schedule obtained by disruption management and the one by waiting machine recovery. Let WAIT be the scheduling strategy of keeping the original schedule fixed, the jobs assigned to the disrupted machine wait until this machine is available and are processed according to the original order. All the other jobs are processed according to the original assignment and order. The LR represents the approach of managing the disruption using the Lagrangian relaxation framework. The objective is the total weighted completion time of all jobs plus the penalty of deviation described before. For each problem instance, let f_{WAIT} be the objective value by the WAIT approach and f_{LR} be the objective value by the LR approach. The relative values of $f'_{\mathrm{WAIT}} = f_{\mathrm{WAIT}}/\min\{f_{\mathrm{WAIT}}, f_{\mathrm{LR}}\}$ and $f'_{\mathrm{LR}} = f_{\mathrm{LR}}/\min\{f_{\mathrm{WAIT}}, f_{\mathrm{LR}}\}$ are

Table 6. Comparison of Objective Function between the Schedule Obtained by Disruption Management and That of Waiting Machine Recovery

M	2		4		6		8		10	
N	WAIT	LR	WAIT	LR	WAIT	LR	WAIT	LR	WAIT	LR
20	1.1773	1.0000	1.0744	1.0000	1.0580	1.0006	1.0622	1.0000	1.0189	1.0064
30	1.0798	1.0000	1.0639	1.0000	1.0566	1.0000	1.0420	1.0006	1.0371	1.0000
40	1.1846	1.0000	1.0835	1.0000	1.0746	1.0018	1.0480	1.0000	1.0293	1.0000
50	1.1784	1.0000	1.0805	1.0000	1.0722	1.0000	1.0529	1.0000	1.0343	1.0002
60	1.1623	1.0000	1.0844	1.0000	1.0476	1.0000	1.0505	1.0000	1.0316	1.0000
70	1.1542	1.0000	1.0829	1.0000	1.0539	1.0000	1.0370	1.0001	1.0413	1.0000
80	1.1735	1.0000	1.0809	1.0000	1.0658	1.0000	1.0622	1.0000	1.0365	1.0000
90	1.1788	1.0000	1.0912	1.0000	1.0489	1.0000	1.0592	1.0000	1.0400	1.0000
100	1.1848	1.0000	1.0833	1.0000	1.0663	1.0000	1.0492	1.0000	1.0459	1.0000
110	1.1823	1.0000	1.0799	1.0000	1.0623	1.0000	1.0460	1.0000	1.0418	1.0000
120	1.1713	1.0000	1.0799	1.0000	1.0535	1.0000	1.0462	1.0000	1.0472	1.0000
130	1.1745	1.0000	1.0818	1.0000	1.0617	1.0000	1.0478	1.0000	1.0384	1.0000
140	1.1819	1.0000	1.0859	1.0000	1.0609	1.0000	1.0521	1.0000	1.0382	1.0000
150	1.1653	1.0000	1.0854	1.0000	1.0541	1.0000	1.0508	1.0000	1.0436	1.0000
160	1.1723	1.0000	1.0804	1.0000	1.0516	1.0000	1.0461	1.0000	1.0401	1.0000
avg	1.1681	**1.0000**	1.0812	**1.0000**	1.0592	**1.0002**	1.0501	**1.0000**	1.0376	**1.0004**

calculated. The average values of f'_{WAIT} and f'_{LR} of 10 runs in each scenario are reported in the respective columns of Table 6.

The result in Table 6 suggests that the LR framework performs better than the approach of leaving the disruption alone on any of the cases discussed herein. We can conclude that the original schedule is no longer satisfying if the disruption situation could not be well-managed. Due to the fact that the performance of the WAIT method will be affected by the duration of breakdown, different lengths of breakdown are designed for 10 groups of instances of the same scenario and the average results are computed. It may be questioned that the WAIT method will be superior to the LR method when the length of breakdown is small enough; however, for such a situation, the experiment indicates that our LR method can provide new schedule with almost the same performance. In the table, for a fixed number of jobs, the duality gap of the disruption management method becomes larger with the decrease of the number of machines. This coincides with practical situation in that the effect of one machine breakdown will not be so serious when the quantity of machines is relative larger. The tighter the resource is, the more important the disruption management will be.

Numerical results in the above tables demonstrate that although the recovery problem is complicated by a variety of factors, the LR algorithm is efficient and effective to obtain satisfying recovery solution in a timely fashion, even if the breakdown continue to the end of the horizon. In detail, the following observations can be made concerning our Lagrangian relaxation algorithm.

(1) α and β are the penalties of virtual earliness and tardiness of original completion time. The results of different values of penalties are acceptable. In fact, the new schedule with job completion time being earlier or later than predefined completion time is undesirable. Therefore, the values of α and β are often set to be the same value to express the same importance of virtual earliness and tardiness in the recovery objective.

(2) When the number of jobs is fixed, as the number of machines increases, the duality gap and computation time decrease. This is consistent with the intuition that for a fixed number of jobs, when the number of machines is larger, resource is less demanded and the problem becomes easier to solve.

(3) For problems with the same numbers of jobs and machines, the length of the disruption strongly affects the performance of the algorithm. When the length of disruption is small enough, the resulted consequence can even be ignored, while the situation is more serious when the length becomes longer.

(4) For all instances, the duality gaps of experiment 1 are smaller than that of experiment 2. This is because the pressure of machine capacity is reduced when the disrupted machines can be available after a period of time and the deviation is dependent on the length of breakdown time. Comparatively, the deviation from the original schedule will be deteriorating when the disrupted machine cannot be used within the horizon. Therefore, for the same original problem, the recovery problem of experiment 1 is more difficult than that of experiment 2.

5. Conclusions

During the execution process of a machine schedule, disruptions often occur when future events do not match expectations. How to deal with the resulted deviations from the original schedule is called disruption management, an emerging field of research. The new problem may be significantly different than the initial one because it sometimes contains new decision variables, new constraints, and a new objective. This paper studies the problem of how to react when an ongoing schedule is disrupted by machine breakdown. On the basis of a general identical machine scheduling model, we develop mathematical model and provide a new way of dealing with the changes to the environment. By extensive computational experiments, we show that our LR algorithm can obtain near-optimal solutions for all problem instances discussed. In addition, our algorithm can complete the recover process within satisfying time, which coincides with the practical situation that, to guarantee production continuity, the updated schedule should be made in a timely manner. Possible future work includes the investigation of more a general scheduling environment such as job shop, hybrid flowshop, and more complex practical scheduling situations.

Acknowledgment

This research is partly supported by National Natural Science Foundation for Distinguished Young Scholars of China (Grant No. 70425003), National 863 High-Tech Research and Development Program of China through approved No. 2006AA04Z174, and National Natural Science Foundation of China (Grant No. 60674084).

Notations

N = total number of jobs to be processed after disruption

p_j = processing time of job j, which is an integer

r_j = ready time of job j, which is an integer

M = current number of machines available

T = time point in scheduling horizon, $t = 1, 2, ..., T$, where T is the length of the horizon:

$$T = \sum_{j=1}^{n} p_j + \max_j r_j$$

α, β = weights of penalty for virtual tardiness and earliness, $\alpha, \beta \geq 0$

$\bar{C}_j$ = completion time of job j in the original schedule, $j = 1, 2, ..., n$, time point ("$\bar{C}_j = t$" means that job j completes at the end of time unit t)

C_j = completion time of job j, $j = 1, 2, ..., n$, time point having the same meaning as $\bar{C}_j$

Decision Variables

$$x_{jt} = \begin{cases} 1 & \text{if job } j \text{ is processed in time unit } t \\ 0 & \text{otherwise} \end{cases},$$

$$j = 1, 2, ..., n; t = 1, 2, ..., T$$

Literature Cited

(1) Love, M.; Sorensen, K. R.; Larsen, J.; Clausen, J. Disruption management for an airline-rescheduling of aircraft. *Lect. Notes Comput. Sci.* **2002**, *2279*, 315–324.

(2) Yu, G.; Argüello, M.; Song, G.; McCowan, S. M.; White, A. A new era for crew recovery at continental airlines. *Interfaces* **2003**, *33* (1), 5–22.

(3) Argüello, M.; Bard, J. F.; Yu, G. A GRASP for aircraft routing in response to groundings and delays. *J. Comb. Optim.* **1997**, *1* (3), 211–228.

(4) Thengvall, B. G.; Bard, J. F.; Yu, G. Multiple fleet aircraft schedule recovery following hubclosures. *Transp. Res. Part A: Policy Pract.* **2001**, *35* (4), 289–308.

(5) Zhu, G.; Bard, J. F.; Yu, G. Disruption management for resource-constrained project scheduling. *J. Oper. Res. Soc.* **2005**, *56*, 365–381.

(6) Vin, J. P.; Ierapetritou, M. G. A new approach for efficient rescheduling of multiproduct batch plants. *Ind. Eng. Chem. Res.* **2000**, *39*, 4228–4238.

(7) Roslöf, J.; Harjunkoski, I.; Björkqvist, J.; Karlsson, S.; Westerlund, T. An MILP-based reordering algorithm for complex industrial scheduling and rescheduling. *Comput. Chem. Eng.* **2001**, *25*, 821–828.

(8) Méndez, C. A.; Cerdá, J. Dynamic scheduling in multiproduct batch plants. *Comput. Chem. Eng.* **2003**, *27*, 1247–1259.

(9) Méndez, C. A.; Cerdá, J. An MILP framework for batch reactive scheduling with limited discrete resources. *Comput. Chem. Eng.* **2004**, *28*, 1059–1068.

(10) Méndez, C. A.; Cerdá, J.; Grossmann, I. E.; Harjunkoski, I.; Fahl, M. State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Comput. Chem. Eng.* **2006**, *30*, 913–946.

(11) Gawiejnowicz, S. Scheduling deteriorating jobs subject to job or machine availability constraints. *Eur. J. Oper. Res.* **2007**, *180*, 472–478.

(12) Kasap, N.; Aytug, H.; Paul, A. Minimizing makespan on a single machine subject to random breakdowns. *Oper. Res. Lett.* **2006**, *34*, 29–36.

(13) Liao, L. W.; Sheen, G.-J. Parallel machine scheduling with machine availability and eligibility constraints. *Eur. J. Oper. Res.* **2008**, *184*, 458–467.

(14) Liao, C. J.; Chen, W. J. Scheduling under machine breakdown in a continuous process industry. *Comput. Oper Res.* **2004**, *31*, 415–428.

(15) Lee, C. Y.; Yu, G. Single machine scheduling under potential disruption. *Oper. Res. Lett.* **2007**, *35* (4), 541–548.

(16) Lee, C. Y.; Leung, J. Y.-T.; Yu, G. Two machine scheduling under disruption with transportation consideration. *J. Sched.* **2006**, *9*, 35–48.

(17) Qi, X. T.; Bard, J. F.; Yu, G. Disruption management for machine scheduling: The case of SPT schedules. *Int. J. Prod. Econ.* **2006**, *103*, 166–184.

(18) Akturk, M. S.; Gorgulu, E. Match-up scheduling undera machine breakdown. *Eur. J. Oper. Res.* **1999**, *112*, 81–97.

(19) Albers, S.; Schmidt, G. Scheduling with unexpected machine breakdowns. *Discrete Appl. Math.* **2001**, *110*, 85–99.

(20) Brown, G. G.; Dell, R. F.; Wood, R. K. Optimization and persistence. *Interfaces* **1997**, *27*, 15–37.

(21) Qi, X. T. Modeling and optimization for disruption management. Ph.D. dissertation, The University of Texas at Austin, 2003.

(22) Luh, P. B.; Yu, D. Q.; Soorapanth, S.; Khibnik, A. I.; Rajamani, R. A Lagrangian relaxation based approach to schedule asset overhaul and repair services. *IEEE Autom. Sci. Eng.* **2005**, *2* (2), 145–157.

(23) Tang, L. X.; Xuan, H.; Liu, J. Y. A new Lagrangian relaxation algorithm for hybrid flowshop scheduling with total weighted completion time. *Comput. Oper Res.* **2006**, *33* (11), 3344–3359.

(24) Gélinas, S.; Soumis, F. Dynamic programming algorithm for single machine scheduling with ready times. *Ann. Oper. Res.* **1997**, *69*, 135–156.

Received for review December 4, 2008

Revised manuscript received April 12, 2009

Accepted June 1, 2009

IE801868F