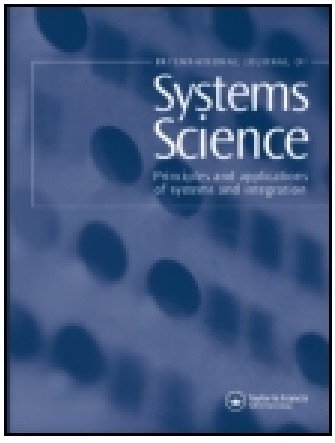

International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

A new Lagrangian Relaxation Algorithm for scheduling dissimilar parallel machines with release dates

Lixin Tang [a] & Yanyan Zhang [a]
[a] Liaoning Key Laboratory of Manufacturing System and Logistics , The Logistics Institute, Northeastern University , Shenyang, 110004, China
Published online: 29 Apr 2010.

To cite this article: Lixin Tang & Yanyan Zhang (2011) A new Lagrangian Relaxation Algorithm for scheduling dissimilar parallel machines with release dates, International Journal of Systems Science, 42:7, 1133-1141, DOI: 10.1080/00207720903308389

To link to this article: http://dx.doi.org/10.1080/00207720903308389

International Journal of Systems Science
Vol. 42, No. 7, July 2011, 1133–1141

Taylor & Francis
Taylor & Francis Group

A new Lagrangian Relaxation Algorithm for scheduling dissimilar parallel machines with release dates

Lixin Tang* and Yanyan Zhang

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

(*Received 7 October 2007; final version received 24 August 2009*)

In this article we investigate the parallel machine scheduling problem with job release dates, focusing on the case that machines are dissimilar with each other. The goal of scheduling is to find an assignment and sequence for a set of jobs so that the total weighted completion time is minimised. This type of production environment is frequently encountered in process industry, such as chemical and steel industries, where the scheduling of jobs with different purposes is an important goal. This article formulates the problem as an integer linear programming model. Because of the dissimilarity of machines, the ordinary job-based decomposition method is no longer applicable, a novel machine-based Lagrangian relaxation algorithm is therefore proposed. Penalty terms associated with violations of coupling constraints are introduced to the objective function by Lagrangian multipliers, which are updated using subgradient optimisation method. For each machine-level subproblem after decomposition, a forward dynamic programming algorithm is designed together with the weighted shortest processing time rule to provide an optimal solution. A heuristics is developed to obtain a feasible schedule from the solution of subproblems to provide an upper bound. Numerical results show that the new approach is computationally effective to handle the addressed problem and provide high quality schedules.

Keywords: Lagrangian relaxation; dissimilar parallel machine; release dates; dynamic programming; machine-based decomposition; heuristics

1. Introduction

Scheduling is concerned with allocating usually limited resources for required tasks over time to achieve certain objectives. It has long been an interesting but challenging research area, and many efforts and improvements have been reported. In process industry, scheduling problems are particularly hard because complex resource and time constraints may have to be simultaneously taken into account. This type of production environment is frequently encountered in chemical industry, where task scheduling have been widely investigated including various plant configurations, such as single machine, parallel machine in single-stage and (hybrid) flow shop, job shop in multiple-stage (Jain and Grossmann 2001; Castro and Grossmann 2006). This article addresses the single-stage scheduling problem, which can be stated that: there are n jobs to be processed on m parallel machines; each job has to be assigned to one machine and each machine can process one job at a time. Jobs need different processing time on different machines. Preemption is usually not allowed, and release/due times may be required. And it is usually assumed that unlimited storage is available. Because multiple machines need to be considered, the solving procedure involves both assignment and sequencing jobs on the machines, which adds more complexity to the problem.

A similar problem can also be found in continuous casting production of steel industry, in which the steel slabs used for downstream hot rolling operation is produced. With the emergence of thin slab casting-rolling (TSCR) technique, the casting speed of continuous caster must be increased to match up with the speed of rolling machines. In continuous casting production usually there are several casters working together in a coordinated manner to provide materials for rolling operation. Casters will have different casting speeds when equipped with different moulds, so the same molten steel will need different casting time when being processed on these machines. In addition, to guarantee the consistency along the production, the release dates may be taken into account when scheduling.

The problem stated above can be viewed as dissimilar parallel machine scheduling, which belongs to unrelated parallel machine scheduling problems but

*Corresponding author. Email: qhjytlx@mail.neu.edu.cn

ISSN 0020–7721 print/ISSN 1464–5319 online

DOI: 10.1080/00207720903308389
http://www.informaworld.com

assumes all machines are different with each other. The goal of the problem is to find an assignment and an ordering of these jobs that minimises the additive objective, subject to release dates. This problem is strong non-deterministic polynomial time (NP)-hard, because its special case the single machine scheduling problem with release dates minimising total weighted completion time has been proved to be NP-hard in the strong sense already (Lenstra, Rinnooy Kan, and Brucker 1977).

Most of research on parallel machine scheduling focuses on the identical case. Nessah, Yalaoui and Chu (2008) designed a branch-and-bound algorithm to minimise total weighted completion time on identical parallel machines with job release dates. Lagrangian relaxation (LR) algorithm has been adapted to a lot of identical parallel machine scheduling problems with job-based decomposition strategy (Luh, Hoitomt, Eric, and Pattipati 1988; Hoitomt, Luh, Max, and Pattipati 1990; Luh and Hoitomt 1993; Ventura and Kim 2003). The job-based decomposition strategy requires that jobs are independent of machines so that relaxed formulation can be expressed as an individual set of each job. Therefore, the LR methods developed for identical machine scheduling are usually inapplicable to the dissimilar case because a job is no longer an independent object. How to decompose the original problem into easier subproblems with independent sets of constraints is the primary task; therefore suitable decomposition strategy must be designed. Therefore, efficient new strategy of decomposition must be explored, which is the purpose of this article.

Regarding unrelated parallel machine scheduling problem, Lenstra, Shmoys, and Tardos (1990) developed a polynomial algorithm which constructs a schedule that is guaranteed to be no longer than twice the optimum. They also provided a polynomial approximation scheme for the case that the number of machines is fixed. Theoretical analysis on both algorithms is presented. Shmoys and Tardos (1993) investigated the problem of scheduling parallel machines with processing time and release date of each job depending on the machines to which it is assigned. They designed an approximation algorithm for minimising a weighted sum of the cost and the maximum job completion time. Hall, Schulz, Shmoys, and Wein (1996) designed and analysed approximation algorithms for NP-hard scheduling problems to minimise the weighted sum of the job completion times under a variety of scheduling environment. Maravelias (2006) proposed a decomposition framework for a dissimilar parallel machine scheduling problem, in which the original problem is decomposed into an assignment and a sequencing subproblem. The hybrid result integrating the solution of the two subproblems exhibits satisfying results with faster speed. Castro and Grossmann (2006) presented a multiple time grid continuous time mixed integer linear programming (MILP) model for the same problem, and their results outperform the former methods. To the best of our knowledge, no result has been reported about applying the LR algorithm to unrelated parallel machine scheduling. Since the LR method can provide a lower bound for NP-hard problem, this article designs a novel algorithm in which the original dissimilar parallel machine scheduling problem is decomposed into machine-level easier subproblems after coupling constraints is relaxed.

In Section 2, the problem statement is given and the mathematical programming model is developed. The description of the LR framework, the solving of the dual problem and the construction of feasible schedule are presented in Section 3. Problem examples 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 statement and mathematical model

2.1. *Problem statement*

In this article, the single-stage scheduling problem under consideration can be stated as follows: n non-preemptive jobs are to be processed on m dissimilar parallel machines. The machines are dissimilar, meaning that p_{jk}, in general, is not equal to p_{jr}, for all jobs j, and all machines $k \neq r$. Each job has a positive weight and a release date after which it can be started. We assume that there are no changeover times and cost. Further, we assume that each job can be processed on any of the parallel machines, and that each machine can process any of all jobs. A job needs different processing time on different machines. A feasible solution is an assignment and processing order of jobs in which each job is started after its release date. The objective is to find a feasible sequence which minimises the sum of weighted completion times.

Mendez, Cerdá, Grossmann, Harjunkoski, and Fahl (2006) summarised the optimisation methods for short-term scheduling of batch processes from various types of modelling aspects, one of which is the unit-specific immediate precedence method. This method has been used by Cerdá, Henning, and Grossmann (1997) to formulate a single-stage batch plant with multiple equipments working in parallel. Pereira Lopes and Valério de Carvalho (2007) developed a branch-and-price algorithm for the unrelated parallel machines with sequence dependent setup times problem. In this article, some modifications are added to the mathematical model by Pereira Lopes and

Valério de Carvalho (2007) to make it suitable for our problem.

Indices:

N Set of jobs, $N=\{1,2,\ldots,n\}$, n is the total number of jobs.
M Set of machines, $M=\{1,2,\ldots,m\}$, m is the available number of machines.
i,j Index of a job, $i,j=1,2,\ldots,n$.
k Index of a machine.

Parameters:

p_{jk} Processing time of job j on machine k.
w_j Weight of job j.
r_j Release date of job j.
C_{jk} Completion time of job j on machine k.

Binary decision variables:

$$x_{ij}^k=\begin{cases}1 & \text{job } j \text{ is processed immediately after job } i \text{ on machine } k,\quad i,j\in N;\ k\in M\\ 0 & \text{otherwise}\end{cases}$$

$$x_{0j}^k=\begin{cases}1 & \text{job } j \text{ is the first job processed on machine } k,\quad j\in N;\ k\in M\\ 0 & \text{otherwise}\end{cases}$$

$$x_{j,n+1}^k=\begin{cases}1 & \text{job } j \text{ is the last job processed on machine } k,\quad j\in N;\ k\in M\\ 0 & \text{otherwise}\end{cases}$$

2.2. *Mathematical model*

$$\underset{\{C_{jk}\}}{\text{Minimize}} \sum_{j\in N}\sum_{k\in M} w_j C_{jk} \tag{1}$$

Subject to:

$$\sum_{k\in M}\sum_{i\in N\cup\{0\}|i\neq j} x_{ij}^k = 1,\quad \forall j\in N \tag{2}$$

$$\sum_{j\in N} x_{0j}^k \le 1,\quad \forall k\in M \tag{3}$$

$$\sum_{i\in N\cup\{0\}|i\neq j} x_{ij}^k = \sum_{i\in N\cup\{n+1\}|i\neq j} x_{ji}^k,\quad \forall j\in N,\ \forall k\in M \tag{4}$$

$$r_j - \sum_{i\in N\cup\{0\}|i\neq j}\sum_{k\in M} x_{ij}^k(C_{jk}-p_{jk})\le 0,\quad \forall j\in N \tag{5}$$

$$C_{jk}\ge \sum_{i\in N\cup\{0\}|i\neq j}(C_{ik}+p_{jk})x_{ij}^k,\quad \forall j\in N,\ \forall k\in M \tag{6}$$

$$x_{ij}^k\in\{0,1\},\quad \forall i\in N\cup\{0\},\ \forall j\in N\cup\{n+1\},\ \forall k\in M. \tag{7}$$

The objective function (1) minimises the total weighted completion time of all jobs. Constraints (2) ensure that each job is processed exactly once, and constraints (3) guarantee that each machine is used at most one time. Constraints (4) enforce the flow conservation that the jobs are correctly ordered within a schedule. Constraints (5) mean that all jobs can be started after they are released, while formula (6) presents the relationship of completion time between two adjacent jobs. Constraints (7) define the value range for the decision variables.

3. Proposed LR approach

An LR is a well-known approach for combinatorial optimisation problems by decomposing the original complex problem into smaller and easier parts through relaxing some coupling constraints. The LR leads to bounds, and often Lagrangian heuristics lead to reasonably good feasible solutions (Weintraub, Church, Murray, and Guignard 2000). Lemaréchal (2003) reviewed the LR method, from both points of view of theory and algorithms. Many successful applications proved that an LR is a general and useful algorithm for combinatorial optimisation problems (Luh, Gou, and Zhang 1998; Luh, Yu, Soorapanth, Khibnik, and Rajamani 2005; Xuan and Tang 2007) by means of decomposition and coordination. The solving procedure of LR can be described as follows: Lagrangian multiplier is introduced to incorporate some 'hard' constraints into objective function to form the LR problem, which can usually be decomposed into smaller and easier subproblems that can be optimally solved. The iterative values of multipliers are obtained according to the degree of violation in the former iteration. By this way, the coordination among the solutions of subproblems is realised. The solution of dual problem of LR can provide a lower bound LB to the original problem, and the modified result from the solution of relaxed problem provides an upper bound UB, the solution quality can be measured by duality gap, the value of $(\text{UB}-\text{LB})/\text{LB}$.

3.1. *LR decomposition*

When LR is employed for constraints decoupling, suitable decomposition strategy needs to be determined. Commonly used methods can be job-based and production stage-based. A job-based decomposition method is useful when there are constraints with regard to the timing of various operations of any given job. Similarly, a production stage-based decomposition method is used when constraints can be divided into independent sets of any given stage. This decomposition procedure solves the problem by scheduling the

production stages one at a time. However, these methods cannot be applied to our problem. The job-based approach cannot be adopted because job's processing time depends on the machine it is assigned to, thus we cannot get the independent set of a single job. And, the stage-based method cannot be used because of the single stage nature of our problem. By observing the structure of the problem, a machine-based decomposition strategy is proposed. The problem is solved by scheduling machines one at a time. To facilitate the decomposition, constraints (2) that couple different machines are relaxed. Further, constraints (5) are relaxed for the design of dynamic programming algorithm. The two relaxations lead to the following LR problem:

(LR)

$$L(u,v)=\min_{C_{jk}} Z_{LR},\ \text{with}$$

$$\begin{aligned}Z_{LR}&=\sum_{j\in N}\sum_{k\in M} w_jC_{jk}+\sum_{j\in N}u_j\left(\sum_{k\in M}\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^k-1\right)\\&\quad+\sum_{j\in N}v_j\Big(r_j-\sum_{i\in N\cup\{0\}|i\neq j}\sum_{k\in M}x_{ij}^k(C_{jk}-p_{jk})\Big)\\&=\sum_{j\in N}\sum_{k\in M} w_jC_{jk}+\sum_{j\in N}u_j\sum_{k\in M}\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^k-\sum_{j\in N}u_j\\&\quad+\sum_{j\in N}v_jr_j-\sum_{j\in N}v_j\sum_{i\in N\cup\{0\}|i\neq j}\sum_{k\in M}x_{ij}^k(C_{jk}-p_{jk})\\&=-\sum_{j\in N}u_j+\sum_{j\in N}v_jr_j+\sum_{j\in N}\sum_{k\in M}\Big(w_jC_{jk}-\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^kv_jC_{jk}\\&\quad+\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^ku_j+\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^kv_jp_{jk}\Big)\end{aligned}\tag{8}$$

subject to constraints (3), (4), (6), (7) and

$$\begin{aligned}&vj\geq 0,\quad j=1,2,\ldots,n,\ u_j\,(j=1,2,\ldots,n)\\&\quad\text{is unrestricted in sign,}\end{aligned}\tag{9}$$

where $\{u_j\}$ and $\{v_j\}$ are the Lagrangian multipliers with regard to constraints (2) and (5), respectively. To find appropriate values of $\{u_j\}$ and $\{v_j\}$, the following Lagrangian dual is constructed.

(LD)

$$\max_{\{u_j,v_j\}} L(u,v),\ \text{with}$$

$$\begin{aligned}L(u,v)=&-\sum_{j\in N}u_j-\sum_{j\in N}v_jd_j+\min_{\{C_{jk}\}}\sum_{j\in N}\sum_{k\in M}\Big(w_jC_{jk}\\&-\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^kv_jC_{jk}+\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^ku_j\\&+\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^kv_jp_{jk}\Big)\end{aligned}\tag{10}$$

subject to constraints (3), (4), (6), (7) and (9).

After being fully decoupled, the original integer programme can be expressed as the sum of the following machine-level subproblems.

(LR_k)

$$\min_{\{C_{jk}\}} L_k(u,v),\ \text{with}$$

$$\begin{aligned}L(u,v)=\sum_{j\in N}\Big(&w_jC_{jk}-\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^kv_jC_{jk}\\&+\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^ku_j+\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^kv_jp_{jk}\Big),\end{aligned}\tag{11}$$

subject to constraints (3), (4), (6), (7) and (9). Therefore, $L(u,v)$ can be rewritten as

$$L(u,v)=-\sum_{j\in N}u_j+\sum_{j\in N}v_jr_j+\min\sum_{k\in M}L_k(u,v).\tag{12}$$

3.2. *Updating Lagrangian multipliers*

Update multipliers $\{u_j\}$

In our algorithm, to solve the dual problem (10), the subgradient optimisation method is used to update multipliers $\{u_j\}$ as follows.

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

where h is the iteration index, α^h is the step size at the hth iteration, $g(u)$ is the subgradient of L at the hth iteration. The subgradient component of $g(u)$ is equal to $(\sum_{k\in M}\sum_{i\in N\cup\{0\}|i\neq j}x_{ij}^k-1)$. The step size α^h is defined as follows.

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

where L^* is the optimal objective value, which is unknown and is usually estimated by the best feasible objective value found so far and L^h is the objective value of the dual function at the hth iteration. Parameter λ used here is initialised with 0.5. It is multiplied by a factor if L^h remains approximately the same value for a fixed number of iterations.

Update multipliers $\{v_j\}$

The same subgradient method is used to update $\{v_j\}$, the only difference is that the subgradient component of $g(v)$ is equal to $(r_j-\sum_{i\in N\cup\{0\}|i\neq j}\sum_{k\in M}x_{ij}^k(C_{jk}-p_{jk}))$.

3.3. *Dynamic programming algorithm for subproblem solving*

In this section, we will introduce in detail the dynamic programming algorithm for solving each subproblem

described by formula (11). In (11) we observe that when v_j is introduced, a new value of weight for each job is obtained as follows.

$$w'_j = w_j - v_j. \tag{15}$$

Obviously, w'_j is a variable weight updated with the new value of multiplier v_j at each iteration. During the iterative procedure of LR algorithm, whenever the dynamic programming is called, the value of w'_j is calculated and a corresponding new job sequence of weighted shortest processing time (WSPT) rule on that machine is obtained. In this way the optimality of each subproblem is guaranteed with respect to the introduced values of Lagrangian multipliers.

The dynamic programming algorithm is constructed based on the property proposed by Belouadah and Potts (1994) that in any optimal schedule, there is no machine idle time between jobs. The algorithm for solving subproblems is an enumerative procedure of all jobs. The stages are the jobs added in the order of increasing index (recall that all jobs have been resorted according to the non-increasing order of w'_j/p_j at the outset of each iteration). Similar researches have been reported (Chen and Warren 1999; van den Akker, Hoogeveen, and van de Velde 1999). Using the same idea that combines dynamic programming and the WSPT rule we construct the dynamic programming algorithm for our problem. Let the set of jobs assigned to one machine be called a subschedule. The algorithm starts with the first job. Let $F(j,t)$ be the minimum cost of a subschedule that processes all jobs in $\{1,2,\ldots,j\}$ exactly once and in which the last job completes at time t. As is shown in Figure 1, at each stage j, job j is either the last job of the sequence, or is not contained in the subschedule. If job j is not included in the subschedule, we must select the best value of $F(j-1,t)$, the solution with respect to the first $j-1$ jobs that finishes at time t. With regard to the second situation, it is considered only if $p_{jk} \le t$. If this is the case, then we add job j to the best solution for the first $j-1$ jobs that finishes at time $t-p_{jk}$; the expression of this solution is $F_{j-1}(t-p_{jk}) + w'_j t + u_j + v_j p_{jk}$. The function F can be recursively calculated by the following relationships:

$$F_j(t) = \begin{cases} \min\{F_{j-1}(t), \quad F_{j-1}(t-p_{jk}) + w'_j t + u_j + v_j p_{jk}\}, \\ \qquad\qquad\qquad \text{if } p_{jk} \le t, \\ F_{j-1}(t), \qquad \text{otherwise} \end{cases} \tag{16}$$

Initially, we set

$$F_j(t) = \begin{cases} 0, & j = 0 \text{ and } t = 0, \\ \infty, & \text{otherwise}, \end{cases} \tag{17}$$

where $j = 1,2,\ldots,n$. As for the determination of time horizon, Belouadah and Potts (1994) provided a common deadline on job completion times for identical case. If job i is completed last in an optimal schedule, let S_i be the start time of job i, then no machine can complete its processing before time S_i. Thus, $S_i \le (\sum_{h=1}^{n} p_h - p_i)/m$, since there is no machine idle time between processing jobs, where p_i is the processing time of job i, m is the number of machines. If $D = \lfloor \sum_{h=1}^{n} p_h/m + (m-1)\max\{p_h\}_{h=1,\ldots,n}/m \rfloor$, it is apparent from the integrality of processing times that $C_i = S_i + p_i \le D$. Thus, D can be regarded as a common deadline on job completion times.

Thus, we can get the common deadline on job completion times for a dissimilar case:

$$D = \left\lfloor \sum_{h=1}^{n} \max_k\{p_{hk}\}/m + (m-1)\max_{k,i}(p_{ik})/m \right\rfloor. \tag{18}$$

With this common deadline, the time horizon t in dynamic programming is set to $t = 0,\ldots,\mathrm{D}$.

This algorithm has been proved (Chen and Warren 1999) to have the worst case complexity $O(nP^k(f))$ since it only takes constant time to compute the value for each of the total n $P^k(f)$ states, where $P^k(f) = \sum_{i=1}^{n} p_{ik}$, k is the current machine.

3.4. *Design of Lagrangian heuristics*

In this section, we introduce the proposed heuristic procedure for dealing with the subproblem solutions, which, when put together, are generally infeasible since machine capacity constraints (2) and constraints (5) have been relaxed. To convert such a solution into a feasible one, we design the following two-stage Lagrangian heuristic procedure.

Stage 1:

Step 1: Put together all the solution of subproblems and obtain a schedule. Check this schedule, count the total times that each job is contained and denote it with *count*(*i*).

Step 2: Keep the positions of all the jobs with *count*(i) $=1$ unchanged then go to step 3.

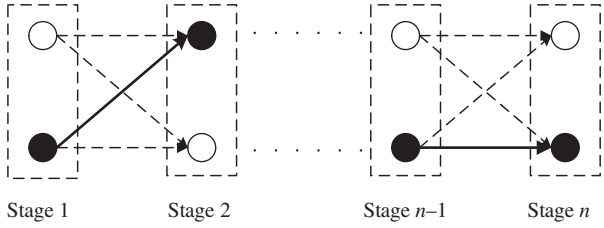

Figure 1. Forward dynamic programming.

Step 3: Let the set of jobs assigned to one machine be called a subschedule. If there exists any job i with $count(i) > 1$, it is implied that this job is contained in more than one subschedule. Keep the position that will lead to minimal objective value, and ensure that feasible position is always prior to infeasible one. Then job i is removed from all other subschedules. Repeat this process until all such jobs have been checked and optimised.

Step 4: When there exists any job i with $count(i) = 0$, that is, this job is not selected in any subschedule, assign it to current idle machine (if there are any). When there is more than one idle machine, assign job i to machine k with the minimal value of p_{ik}, let $count(i) = 1$.

Step 5: Step 6. Try to insert an unselected job in each subschedule, keep the best and feasible results. If all unselected jobs have been inserted, stage 1 ends, otherwise assign it to the machine with the smallest workload at that moment.

Stage 2:

In stage 2, the heuristic algorithm proposed by Smith (1956) is adapted to sequence the jobs assigned to the same machine. The algorithm starts from the first position by choosing job j which has the largest value of w_j/p_j among the set of jobs whose release dates are at least as small as the completion time of the immediately preceding job. If no such job exists, job j is the one with the earliest release date. Then job j is placed in the first position and removed from the problem and the procedure repeats until all jobs are sequenced. Formally, the algorithm for dissimilar case can be stated as follows.

Step 1: Let I be the set of all jobs, let l be corresponding position on machine k, initially, $l = 1$; let S_l be the start time of the job assigned to position l.

Step 2: Find the set $I_r = \{j \mid j \in I, r_j \leq S_l\}$, if set I_r is not empty, find a job j in I_r having the largest value of w_j/p_{jk}, go to step 3; otherwise find a job j with the earliest release date in I, go to step 3.

Step 3: Sequence job j in position l, set $l = l + 1$, set $I = I - \{j\}$. If $I = 0$, then stop; otherwise go to Step 2.

3.5. *Algorithm framework*

To sum up, the whole procedures of LR algorithm can be illustrated as shown in Figure 2.

The parameters need to be initialised include Lagrangian multipliers $\{v_j\}$ and $\{u_j\}$. Initially, all elements of the multipliers are set to be 0.

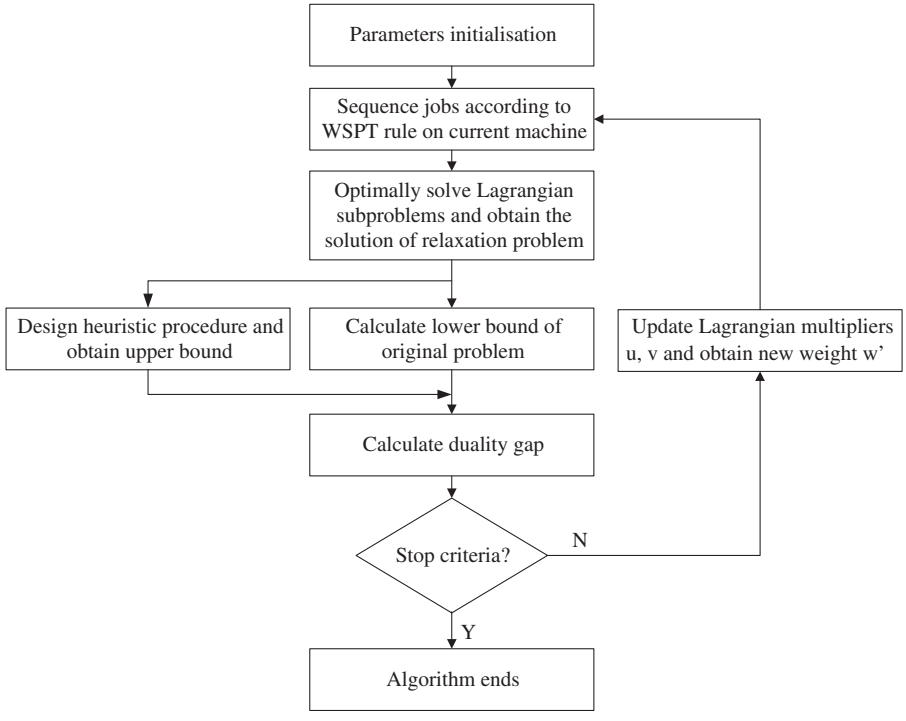

Figure 2. Framework illustration of the proposed LR algorithm.

4. Computational results

The test problems were generated as follows using the method proposed by Gélinas and Soumis (1997) to guarantee data rationality. The algorithm is tested on problems with up to 100 jobs and 10 machines. The number of jobs $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, the number of machines $m \in \{2, 4, 6, 8, 10\}$. For each job j a processing time p_{jk} is selected as an integer from the uniform distribution [20, 50] and a real weight w_j is selected in the uniform distribution [1, 5]. The relative smaller value of weight is set because a larger value will submerge the effect of multiplier v_i, leading to the same WSPT sequence at any iteration. Reasonable value of release dates are generated by selecting jobs at random and then scheduling them at the earliest possible time t_i. This schedule is used to generate problems with integer release dates $r_i = t_i - \mathrm{U}[1, 40]$. A total of 50 problem types are generated, each having 10 test examples.

To evaluate the performance of our method, the examples were implemented by language C++ on a PC with Pentium-IV (2.40 GHz) CPU with 512 CDRAM using the Windows 2000 operating system. For each problem type, 10 test examples are implemented. The algorithm ends when the maximum number of iterations (500) is reached or the predefined value of duality gap (0.5%) has been achieved. These are the typical values of most LR applications. In practice, with respect to various problems, the maximum number of iterations may be set to be larger than 500, but usually smaller than 1000. Too much iteration in fact cannot provide further improvement, just time consuming. This is the same for the set of the stop criteria of duality gap. The average results of 10 data are presented in Tables 1 and 2 and Figures 3 and 4. Table 1 and Figure 3 show the average duality gap of all test examples, where duality gap is calculated by: gap $= (UB - LB)/LB \times 100\%$.

From the results presented in Table 1 and Figure 3, the following observations can be made:

(1) For all test examples, the duality gaps, on average, are no more than 5%, indicating that better ranges of optimal solutions have been found for this NP-hard problem. Naturally, the

Table 1. Results of duality gap (unit: %).

		M				
Problem	N	2	4	6	8	10
1	10	3.96	1.89	1.90	5.45	3.66
2	20	2.97	3.30	3.52	2.75	4.37
3	30	2.74	3.00	5.08	8.66	8.52
4	40	2.47	2.70	4.36	6.23	4.84
5	50	2.09	2.46	2.99	3.84	5.47
6	60	1.38	2.52	2.74	4.94	4.14
7	70	1.77	2.27	3.16	2.73	3.51
8	80	1.53	2.14	2.56	3.24	3.64
9	90	1.53	2.19	2.51	3.96	3.80
10	100	1.32	1.77	2.62	3.19	3.50
Average		2.18	2.42	3.14	4.50	4.55

Table 2. Results of running time (unit: s).

		M				
Problem	N	2	4	6	8	10
1	10	0.14	0.13	0.15	0.15	0.24
2	20	0.52	0.73	0.83	0.96	1.09
3	30	1.00	1.41	1.84	4.23	2.81
4	40	2.20	3.05	4.41	5.68	10.57
5	50	2.75	3.91	6.46	5.68	10.69
6	60	3.18	6.81	9.83	9.63	11.71
7	70	7.67	7.09	12.63	12.59	20.76
8	80	10.84	11.92	14.92	15.96	26.85
9	90	13.78	13.84	33.64	25.54	63.20
10	100	17.59	17.85	22.33	28.83	33.92
Average		5.97	6.67	10.70	10.93	18.18

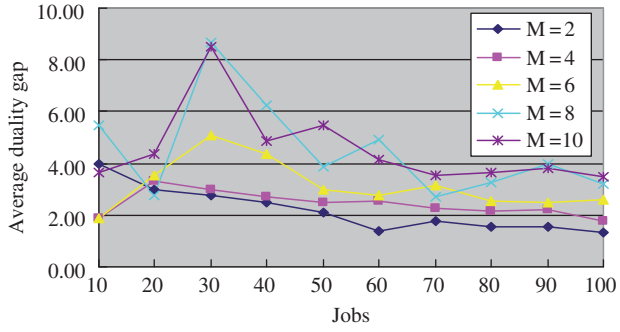

Figure 3. Comparison of average duality gap for different number of machines.

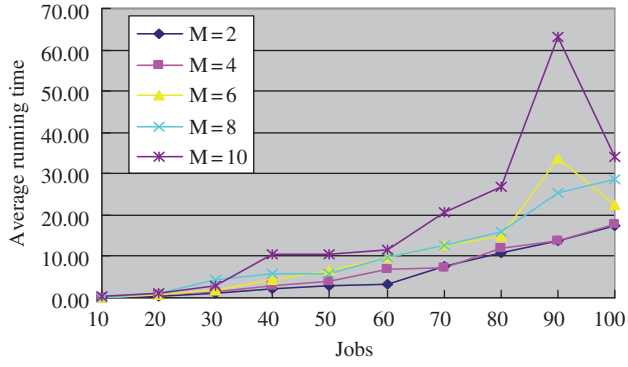

Figure 4. Comparison of average running time for different number of machines.

average duality gap increases as the number of machines increase.

(2) For fixed number of jobs, the duality gap increases approximately with the number of machines, which can be seen from the data in each row of Table 1. More intuitional trends can be observed in Figure 3.

In Table 2 and Figure 4 the average computational time is counted in seconds. From the results we observed that:

(1) Our algorithm solves problems in a reasonable computational time with the longest time of about 60 min.

(2) For fixed number of jobs, the time consumption increases as the number of machines increase; for fixed number of machines, the time consumption increases as the number of jobs increase.

To sum up, the proposed LR algorithm with machine-based decomposition strategy is suitable for dealing with dissimilar parallel machine scheduling problem, which cannot be solved by the commonly used decomposition method. The test results demonstrate the effectiveness and the efficiency of the method.

5. Conclusions

In this work, we developed a novel LR algorithm for the problem of scheduling a set of independent jobs with release dates, on dissimilar parallel machines to minimise the total weighted completion time. The whole problem is solved by using the LR algorithm with the subgradient optimisation framework. Respective penalties have been added to the violations of coupling constraints to relax the hard original problem into machine-level subproblems. The dynamic programming algorithm is designed to obtain an optimal solution of each subproblem. The algorithm solves problems with up to 10 machines and 100 jobs in a reasonable computational time. The testing results demonstrate that the new method generates acceptable schedules for all instances within satisfying computational time.

Acknowledgement

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).

Notes on contributors

Lixin Tang received the B.Eng. degree in industrial automation, the M.Eng. degree in systems engineering and the Ph.D. degree in control theory and application from Northeastern University, Shenyang, China, in 1988, 1991 and 1996, respectively. Currently, he is a Chair Professor of 'the Cheung Kong Scholars Programme of China', and the Director of both the Liaoning Key Laboratory of Manufacturing System and Logistics, and the Logistics Institute at Northeastern University, China. His research interests include operations planning, production scheduling, logistics and supply chain management and combinational optimization. He has published a monograph and more than 50 papers in international journals, such as *Computers & Operations Research*, *Computers and Industrial Engineering*, *European Journal of Operational Research*, *IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Control System Technology*, *Industrial & Engineering Chemistry Research*, *International Journal of Management Science*, *International Journal of Production Economics*, *International Journal of Production Research*, *International Transactions in Operational Research* and *Journal of the Operational Research Society*. He has received National Natural Science Foundation for Distinguished Young Scholars of China, State Youth Science and Technology Award of China, Outstanding Young Faculty Award Program of the Ministry of Education and Fok Ying Tung Education-Foundation Reward.

Yanyan Zhang received the B.Eng. degree in Automation and Instrument, the Ph.D. degree in logistics from Northeastern University, Shenyang, China, in 1998 and 2008 respectively. Her research interests include production scheduling, disruption management, power scheduling, energy allocation, unit commitment with pricing and combinational optimisation. She has published nine papers in international journals and conference.

References

Belouadah, H., and Potts, C.N. (1994), 'Scheduling Identical Parallel Machines to Minimize Total Weighted Completion Time', *Discrete Applied Mathematics*, 48, 201–208.

Castro, P.M., and Grossmann, I.E. (2006), 'An Efficient MILP Model for the Short-Term Scheduling of Single Stage Batch Plants', *Computers and Chemical Engineering*, 30, 1003–1018.

Cerdá, J., Henning, G.P., and Grossmann, I.E. (1997), 'A mixed-Integer Linear Programming Model for Short-Term Scheduling of Single-Stage Multiproduct Batch Plants with Parallel Lines', *Industrial and Engineering Chemistry Research*, 36, 695–707.

Chen, Z.L., and Warren, B.P. (1999), 'Solving Parallel Machine scheduling Problems by Column Generation', *INFORMS Journal on Computing*, 11, 78–94.

Gélinas, S., and Soumis, F. (1997), 'A Dynamic Programming Algorithm for Single Machine Scheduling with Ready Times', *Annals of Operations Research*, 69, 135–156.

Hall, L.A., Schulz, A.S., Shmoys, D.B., and Wein, J. (1996), 'Scheduling to Minimize Average Completion Time: Off-line and On-line Approximation Algorithms', *Mathematics of Operations Research*, 22, 513–544.

Hoitomt, D.J., Luh, P.B., Max, E., and Pattipati, K.R. (1990), 'Scheduling Jobs with Simple Precedence Constraints on Parallel Machines', *IEEE Control System Magazine*, 10, 34–40.

Jain, V., and Grossmann, I.E. (2001), 'Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems', *INFORMS Journal on Computing*, 13, 258–276.

Lemaréchal, C. (2003), 'The Omnipresence of Lagrange', *4OR*, 1, 7–25.

Lenstra, J.K., Rinnooy Kan, A.H.G., and Brucker, P. (1977), 'Complexity of Machine Scheduling Problems', *Annals of Discrete Mathematics*, 1, 343–362.

Lenstra, J.K., Shmoys, D.B., and Tardos, É. (1990), 'Approximation Algorithms for Scheduling Unrelated Parallel Machines', *Mathematical Programming*, 46, 259–271.

Luh, P.B., and Hoitomt, D.J. (1993), 'Scheduling of Manufacturing Systems Using the Lagrangian Relaxation Technique', *IEEE Transaction on Automatic Control*, 38, 1066–1079.

Luh, P.B., Gou, L., and Zhang, Y. (1998), 'Job Shop Scheduling with Group-Dependent Setups, Finite Butters, and Long Time Horizon', *Annals of Operations Research*, 76, 233–259.

Luh, P.B., Hoitomt, D.J., Eric, M., and Pattipati, K.R. (1988), 'Parallel Machine Scheduling Using Lagrangian Relaxation', *Proceedings of the International Conference on Computer Integrated Manufacturing, 1988*, 244–248.

Luh, P.B., Yu, D., Soorapanth, S., Khibnik, A.I., and Rajamani, R. (2005), 'A Lagrangian Relaxation Based Approach to Schedule Asset Overhaul and Repair Services', *IEEE Transactions on Automation Science and Engineering*, 2, 145–157.

Maravelias, C.T. (2006), 'A Decomposition Framework for the Scheduling of Single- and Multi-Stage Processes', *Computers and Chemical Engineering*, 30, 407–420.

Mendez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., and Fahl, M. (2006), 'State-of-the-Art Review of Optimization Methods for Short-Term Scheduling of Batch Processes', *Computers and Chemical Engineering*, 30, 913–946.

Nessah, R., Yalaoui, F., and Chu, C. (2008), 'A Branch-and-Bound Algorithm to Minimize Total Weighted Completion Time on Identical Parallel Machines with Job Release Dates', *Computers and Operations Research*, 35, 1176–1190.

Pereira Lopes, M.J., and Valério de Carvalho, J.M. (2007), 'A Branch-and-Price Algorithm for Scheduling Parallel Machines with Sequence Dependent Setup Times', *European Journal of Operational Research*, 176, 1508–1527.

Shmoys, D.B., and Tardos, E. (1993), 'An Approximation Algorithm for the Generalized Assignment Problem', *Mathematical Programming*, 62, 461–474.

Smith, W.E. (1956), 'Various Optimizers for Single-Stage Production', *Naval Research Logistics*, 3, 59–66.

van den Akker, J.M., Hoogeveen, J.A., and van de Velde, S.L. (1999), 'Parallel Machine Scheduling by Column Generation', *Operations Research*, 47, 862–872.

Ventura, J.A., and Kim, D. (2003), 'Parallel Machine Scheduling with Earliness-Tardiness Penalties and Additional Resource Constraints', *Computers and Operations Research*, 30, 1945–1958.

Weintraub, A., Church, R.L., Murray, A.T., and Guignard, M. (2000), 'Forest Management Models and Combinatorial Algorithms: Analysis of State of the Art', *Annals of Operations Research*, 96, 271–285.

Xuan, H., and Tang, L.X. (2007), 'Scheduling a Hybrid Flowshop with Batch Production at the Last Stage', *Computers and Operations Research*, 34, 2718–2733.