Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Tabu search based resource allocation in radiological examination process execution*

Chun-hua HE

College of Medical Engineering Technology, Xinjiang Medical University, Urumqi 830011, China

E-mail: 460938212@qq.com

Received Dec. 11, 2016; Revision accepted Jan. 23, 2017; Crosschecked Mar. 15, 2018

Abstract: Efficient resource scheduling and allocation in radiological examination process (REP) execution is a key requirement to improve patient throughput and radiological resource utilization and to manage unexpected events that occur when resource scheduling and allocation decisions change due to clinical needs. In this paper, a Tabu search based approach is presented to solve the resource scheduling and allocation problems in REP execution. The primary objective of the approach is to minimize a weighted sum of average examination flow time, average idle time of the resources, and delays. Unexpected events, i.e., emergent or absent examinations, are also considered. For certain parameter combinations, the optimal solution of radiological resource scheduling and allocation is found, while considering the limitations such as routing and resource constraints. Simulations in the application case are performed. Results show that the proposed approach makes efficient use of radiological resource capacity and improves the patient throughput in REP execution.

Key words: Radiological examination process (REP); Resource scheduling and allocation; Tabu search
https://doi.org/10.1631/FITEE.1601802 **CLC number:** TP391; O226

1 Introduction

The radiological examination process (REP), e.g., X-ray computer tomography (CT) scans, initiated by a request for a patient radiological examination procedure, requires the participation and cooperation of many radiological resources, i.e., medical equipment, physicians, and technicians. Resources are important indicators of REP performance. In REP execution, a resource indicates an actor or agent that carries out REP activities. From this point of view, REP is essentially a resource-driven process (Huang et al., 2011b).

In a radiology department, the radiological resources are significantly burdened by REP activities. Thus, radiological resource scheduling and allocation (RRSA) in REP execution is a vital task. Usually, an RRSA solution is obtained manually by the department managers. In practice, this often leads to problems of allocating too much or too little work to resources, and even allocating work to inappropriate resources. As a consequence, many errors and unwanted effects occur. For example, a patient may have to wait because resources are not available due to bad scheduling; REP execution may be postponed, canceled, or require latency time; subsequent appointments may then have to be rescheduled. Thus, the optimal RRSA solution, as the process of constructing workable timetables for process participants, becomes crucial.

Efficient decisions on RRSA are complex and dynamic tasks due to routing and resource constraints (Duftschmid et al., 2002; Huang et al., 2011a). Particularly, limitations in health-care funding require managers of the radiology department find effective ways to use resources. Because patient stays at the radiology department affect medical costs, the

* Project supported by the National Natural Science Foundation of China (No. 61562088)
ORCID: Chun-hua HE, http://orcid.org/0000-0002-6517-7208

minimization of patient examination flow time to provide a high patient throughput is of great importance (Huang et al., 2011b). On the other hand, the running costs of some kinds of resources (e.g., medical equipment) are very high when they are idle and can be turned off on time. Thus, the objective of RRSA is a weighted sum of the average examination flow time of patients, the average idleness of resources during the length-of-time horizon, and delays.

The RRSA problem is similar to the resource-constrained job shop scheduling problem (Verhoeven, 1998; Mika et al., 2008; Vilcot and Billaut, 2008; Huang et al., 2010). There are many approaches to resource-constrained job shop scheduling. The most popular approaches are analytical and heuristic methods. Analytical methods use mathematical programming, such as linear programming or dynamic programming, to solve problems (Elmaghraby, 1993). Generally, mathematical models are difficult to create and require much computation effort. They are not suitable for dealing with the RRSA problem, because the size of REP cases is large and unexpected events always occur in practice. Recently, some heuristic methods have been developed because they are simple and easy to use (Brah and Loo, 1999; Kubzin et al., 2009). Although these methods could produce good solutions, they do not guarantee optimality and are proven to be problem-dependent. Furthermore, these approaches are generally targeted at a specific application area and cannot be easily transferred to clinical scenarios (Huang et al., 2011b).

In this study, we investigate RRSA at the radiology department in the Chinese Huzhou Hospital. In recent years, a process-aware radiology information system has been implemented and the whole logistical process around radiological examination has been improved substantially in the Chinese Huzhou Hospital (Zhang et al., 2009). However, the actual RRSA is still created using simple rules (i.e., first in, first out). There is often a lack of overview on how these low-level scheduling and allocation decisions influence the overall performance.

In this study, we address a Tabu search (TS) based approach to solve the RRSA problem with additional constraints related to the examination type, resource capacity, unexpected events (i.e., emergent or absent examinations), etc. The objective is the joint minimization of average examination flow time, average resource idle time, and delay.

2 Radiological examination process

The radiological examination process is typically structured, yet with slight flexibility. In this study, we take the case of CT scans at the radiology department in the Chinese Huzhou Hospital. The REP for the CT scan examination process is literally central in the clinical pathway of many patients. It covers a set of medical transactions including patient registration, examination scheduling and control, examination report generation and transcription, transfer of patient data to examination facilities (modalities), and storage, printing, archiving, or forwarding of the generated image data, and so on.

Fig. 1 illustrates a basic CT scan examination process model, which is expressed in terms of a workflow net (Salimifard and Wright, 2001), i.e., a Petri net describing the life cycle of a case. The process starts with the ‘receive an examination order’ activity (A). A physician of clinical departments sends a CT scan examination order for a patient to the radiology department. After receiving the order, patient information, including the radiology order with multiple requested exam procedures, is registered and an examination appointment is arranged (B). The patient arrives at the appointment time to undergo a radiology examination according to the requested exam

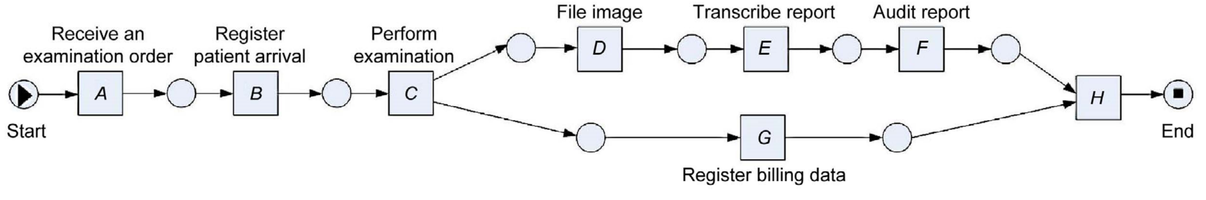

Fig. 1 Radiological CT scan examination process model
The model is expressed in terms of a workflow net (a Petri net describing the life cycle of a case), where rectangles represent resources, circles represent activities, and arrows represent transitions from one activity to another

procedures (C). The appointed time denotes an examination case at the earliest possible starting time. After that, images related to the patient are processed and stored to the archive (D). Then radiologists query images and create or edit report files (E), which document the radiology examination result and include the interpretation and the impressions of the radiologist. Report files should be audited by another senior radiologist (F). In parallel, the examination bill is processed (G). After that, report files are submitted to the server (H) and the case is closed.

REPs like the one described previously have two important characteristics. First, they use various kinds of resources: equipment resources (CT machines, etc.), logistics resources (radiology information system (RIS), picture archiving and communication system (PACS), hospital information system (HIS), etc.), and human resources (technicians, radiologists, etc.). Second, although the number of examination cases is fixed and known in advance in a fixed-period time horizon (i.e., from 8:00 AM to 12:00 AM) with respect to medical equipment capacity constraints, workforce, and patient demands, unpredictable events always occur (e.g., emergent examination requests without an appointment or missed examinations), which is an additional element of difficulty in defining an efficient RRSA solution. Furthermore, unpredictable cases (e.g., emergent examinations) should be dealt with online as soon as they occur by revising the current scheduled activities. In this context, makespan minimization, as a traditional measurement method in industrial process scheduling, is not an economically relevant measure of schedule performance. The objective of using radiological resources efficiently to minimize average examination flow time with patient priority consideration and to minimize resource idle time and delays, is established instead.

3 Problem formulation

The RRSA problem takes into account the assignment of activities to appropriate resources in REP execution. An optimal RRSA solution should satisfy both routing and resource constraints and should be minimal (or maximal) with respect to the given objective function.

In this study, we present a basic procedure to solve the RRSA problem. First, we make the following assumptions:

1. In a time horizon (i.e., from 8:00 AM to 12:00 AM), the number of REP cases is the sum of the number of appointment cases and unpredictable emergent cases minus the number of missed examinations. The number of appointments is known in advance. The number of emergent cases follows a Poisson distribution with parameter λ. Missed examinations occur frequently in practice. We assume that each appointment examination has a probability ρ of being missed. Here, ρ is the same for all appointments and those examinations are independent.

2. Each REP case consists of a set of activities in a prescribed order. All activities considered are split into G groups according to their examination types.

3. Each REP case cannot start before its appointment time (or arrival time for emergent cases), and its processing should not exceed its scheduled duration.

4. There are a set of resources. For each type of activity, a subset of resources are qualified to perform it, and the execution time is fixed and known in advance.

5. Each activity can be performed without interruption by one resource.

Based on the above assumptions, we give the indices, parameters, and variables in Table 1. The objective of the RRSA problem is to find a schedule that has a minimum weighted sum of the average examination flow time of patients, the idleness of resources during the time horizon, and delays:

$$\begin{aligned}\min_{x_{ijk},y_{ijg},z_{gr}} \ & \alpha_{\mathrm{F}}\frac{1}{I}\sum_{i=1}^{I}(\overline{\tau}_i-\underline{\tau}_i)\\ &+\alpha_{\mathrm{I}}\frac{1}{R}\sum_{k=1}^{R}\sum_{g=1}^{G}\sum_{i=1}^{I}\sum_{j=1}^{J}(s_{ij}-c_{ij'})x_{ijk}\,y_{ijg}\,z_{gr}\\ &+\alpha_{\mathrm{T}}\max\left\{\max\{\overline{\tau}_i \mid \forall i\in I\}-T,0\right\}\end{aligned} \tag{1}$$

subject to

$$\sum_{k=1}^{R}\sum_{i=1}^{I}\sum_{j=1}^{J}x_{ijk}=1, \tag{2}$$

$$1\le\sum_{r=1}^{R}z_{gr}\le R, \tag{3}$$

Table 1 Indices, parameters, and variables of the RRSA problem

Type	Label	Description
Index	i	REP case
	j	Activity
	g	Activity group
	k	Resource
Parameter	I	Size of examination cases, $I=I_{\text{app}}(1-\rho)+E\{I_{\text{emr}}\}$, where I_{app} is the number of appointed examinations, ρ the probability of a missed appointment, $E\{I_{\text{emr}}\}=\lambda T$ the expectation value of the number of emergent examinations in time horizon T, and λ the rate parameter of the number of emergent cases, which follows the Poisson distribution
	J	Size of activities of each case
	G	Size of activity groups
	A_i	The set of all activities to cover for each case i, $A_i=\{a_{i1}, a_{i2}, ..., a_{iJ}\}$
	$\text{Pre}(a_{ij})$	The set of precedent activities of a_{ij}
	$\text{Suc}(a_{ij})$	The set of successor activities of a_{ij}
	R	The set of resources
	R_{ij}	The set of resources that can be used by activity j for case i
	r_k	Resource k
	$\underline{\tau}_i$	The appointment time (or arrival time for an emergent case) of case i
	$\overline{\tau}_i$	The completion time of case i
	s_{ij}	Start time of activity a_{ij}
	c_{ij}	Completion time of activity a_{ij}
Variable	x_{ijk}	A binary variable that takes value 1 if a_{ij} is executed by resource r_k and 0 otherwise
	y_{ijg}	A binary variable that takes value 1 if a_{ij} is from group g and 0 otherwise
	z_{gr}	A binary variable that takes value 1 if resource r is qualified to perform activities from group g and 0 otherwise

$$\sum_{g=1}^{G} y_{ijg} = 1, \tag{4}$$

$$c_{ij'} < s_{ij},\ \forall a_{ij'} \in \text{Pre}(a_{ij}), \tag{5}$$

$$c_{ij} < s_{ij'},\ \forall a_{ij'} \in \text{Suc}(a_{ij}), \tag{6}$$

$$\overline{\tau}_i = \max(c_{ij}), \tag{7}$$

where j' denotes activities prior activity j, and α_{F}, α_{I}, and α_{T} are weights of average examination flow time with the prior consideration, average resource idle time, and delay, respectively. Constraint (2) guarantees that each activity is performed by one resource. Constraint (3) guarantees that each type of activity can be performed by a set of qualified resources. Constraint (4) guarantees that each activity has a certain type, and constraints (5) and (6) represent the predetermined sequence of activities performed on the same cases. Constraint (7) determines the completion time of a case based on the completion time of the last activity.

4 Tabu search

In this section, a TS-based approach is proposed to solve the RRSA problem. TS, originally developed by Glover (1986), is a local search technique.

The TS technique is based on an iterative procedure, the 'neighborhood search method', for finding, in a finite set Sol of feasible solutions, a solution $\text{sol}^* \in \text{Sol}$ that minimizes a real-valued objective function $f(\cdot)$. Neighborhood search methods are iterative procedures in which a neighborhood $N(\text{sol})$ is defined for each solution $\text{sol} \in \text{Sol}$ and the next solution is searched among the solutions in $N(\text{sol})$, obtained by a predefined partial modification of sol, usually called a move. Starting from an initial feasible solution, the neighborhood $N(\text{sol})$ of the current solution sol is examined and the solution sol with usually the best objective function value is chosen as the next solution. The fact that movement from solution sol to another solution sol′ is allowed even if $f(\text{sol}')>f(\text{sol})$ helps escape from local optima.

To prevent cycling, a structure called a Tabu list (TL) is introduced to prevent returning to a solution visited in the last |TL| (length of TL) iterations. The TS process stops when the solution is close enough to the lower bound of the objective function value, if known. Otherwise, it stops when no improvement occurs over the best solution for a given number of iterations or the time limit runs out.

4.1 Solution representation

In this study, a solution is represented by a disjunctive graph $G=(N, A_T, A_R)$, where $N=\{a_{ij}, 1\leq i\leq J\}\cup\{s, c\}$ is the set of nodes with s being the source and c the sink node, A_T the arcs corresponding to routing constraints among activities, and A_R the arcs corresponding to resource constraints.

Arcs A_T cannot be modified and are the same for all possible solutions. There is an arc between s and all the initial activities of case i of length L_i, an arc between activity a_{ij} and activity $a_{ij'}$ of length L_{ijk} if a routing constraint imposes that a_{ij}, assigned to resource r_k, has to precede $a_{ij'}$ and, finally, an arc between activity a_{iJ} and c of length L_{iJk} if activity a_{iJ} is assigned to resource r_k.

On the other hand, arcs A_R will be modified by the procedures. If activity a_{ij} precedes activity $a_{i'j'}$ by resource r_k, there is an arc between a_{ij} and $a_{i'j'}$ of length $L_{ijk}+S_{ijki'j'}$. More precisely, $A_R=\cup_{k=1}^{R} A_R^k$,

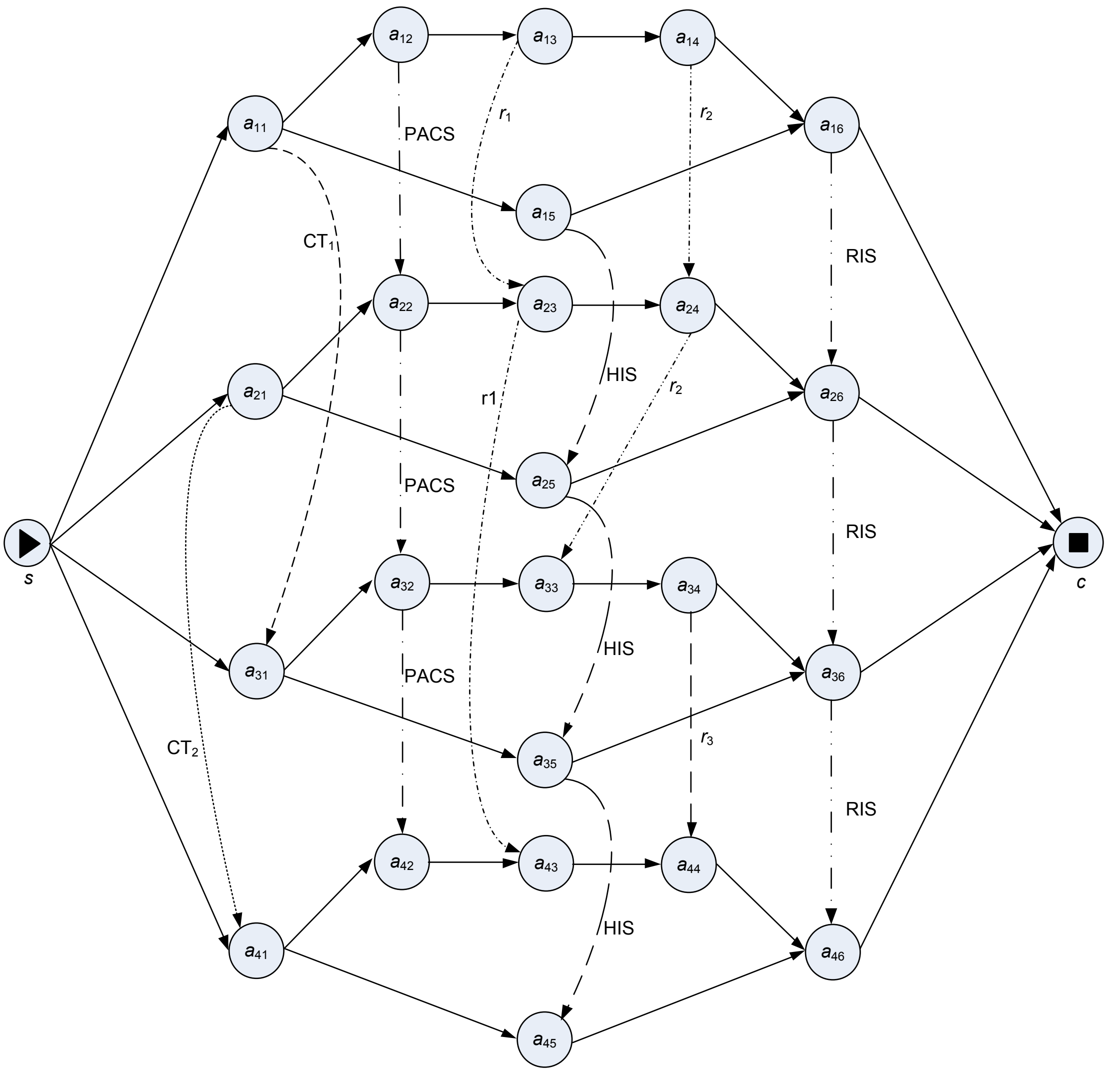

Fig. 2 Solution representation based on a disjunctive graph for four radiological CT scan examination process cases
Circles represent activities, dotted and dashed arrows represent different resources, and solid arrows represent transitions between activities

where A_R^k is the subset of the disjunctive arc of A_R related to resource k.

The paths between s and c give the solution to the RRSA problem. Fig. 2 illustrates a representation with four CT scan examination process cases and the following sequences on resources: (a_{11}, a_{31}) on resource CT_1, (a_{21}, a_{41}) on resource CT_2, (a_{12}, a_{22}, a_{32}, and a_{42}) on resource PACS, (a_{13}, a_{23}, and a_{43}) on resource r_1, (a_{14}, a_{24}, and a_{33}) on resource r_2, (a_{34}, a_{44}) on resource r_3, (a_{15}, a_{25}, a_{35}, and a_{45}) on resource HIS, and (a_{16}, a_{26}, a_{36}, and a_{46}) on resource RIS. Note that these cases follow part of the CT scan examination process model in this example.

4.2 Initial solution

Inspired by Negenman (2001), in this study, we find an initial solution by a heuristic to balance the resource workload. First, the activities are sorted in nondecreasing order of $|R_{ij}|$ and by nonincreasing order of $|R_{ij}|\sum_{k\in R_{ij}} L_{ijk}$ to break ties. The workload of one resource is defined by the sum of the processing times of the assigned activities. Resources are sorted in nondecreasing order of their workload. For each activity a_{ij}, the first resource belonging to R_{ij} is assigned to the activity, and the workload of this resource is updated, as well as the sorting of the resources. The process iterates until each activity is assigned to one resource.

Furthermore, we denote by $\Im$ the set of candidate activities. At the beginning, $\Im$ contains all activities without a predecessor. Candidate activities are sorted in nondecreasing order of their arrival time. Slack of activity a_{ij} is the difference between the scheduled completion time of process case i and the completion time of this case if the remaining activities after a_{ij} are processed without idle time (for each successor $a_{ij'}$, we consider $\max_{k\in R_{ij'}} L_{ij'k}$). The release time of an activity a_{ij} is the maximum between the time when its resource is free for processing activity a_{ij} and the maximum completion time of all predecessors of a_{ij}. The first candidate activity is scheduled, the release dates are updated, $\Im$ is updated, and the process iterates while $\Im\neq\varnothing$.

4.3 Neighbor generation

A neighbor is obtained by moving an activity in a disjunctive graph. This means deleting and adding arcs corresponding to resource constraints (routing constraints are not changed). Note that a feasible move is a move of which activity is assigned to a resource, where it can be processed, such that no cycle is generated in the graph (Negenman, 2001). For each activity and each feasible move, a neighbor is generated. A neighbor is evaluated by its path length. The best neighbor is the neighbor that is not in the Tabu list and with a minimum path length.

There are two kinds of basic moves defined in our study:

1. Swapping move: Swap two activities that are performed by the same resource. Suppose there is a set of activities in a resource work list in sequence: (..., a_{i-1}, a_i, a_{i+1}, ..., a_{j-1}, a_j, a_{j+1}, ...). The move is to swap the positions of a_i and a_j. Then, it needs to first delete arcs (a_{i-1}, a_i), (a_i, a_{i+1}), (a_{j-1}, a_j), and (a_j, a_{j+1}) and then add arcs (a_{i-1}, a_j), (a_j, a_{i+1}), (a_{j-1}, a_i), and (a_i, a_{j+1}).

2. Insertion move: Remove an activity from the work list of one resource and insert it into the work list of another resource. Note that both resources are qualified to perform that activity. Suppose the work lists of the two resources are (…, a_{i-1}, a_i, a_{i+1}, …) and (…, a_{j-1}, a_j, …). The move is to remove activity a_i from the first work list and insert a_i between a_{j-1} and a_j in the second work list.

The above two moves may generate a cycle in the graph and result in a 'deadlock' of scheduling. Fig. 3 shows two examples of deadlock after a move. This problem has been studied in a slightly more general form in Negenman (2001), where a feasible move is defined if it does not create a cycle in the resulting graph. Our approach does not perform moves that lead to cyclic graphs. To this aim, an acyclicity test is preliminarily checked before each move and the move is removed from the neighborhood if it leads to a cycle. In this study, we use the conditions of the following Theorem 1 to perform the acyclicity check.

Suppose x and y are two activities such that there is a directed path from x to y in G, denoted by $x\prec y$. Let Ω_k be the sequence of activities that are allocated to resource $r_k\in R$; let $\Omega:=\cup_{r_k\in R}\Omega_k$ be the set of all sequences. When swapping the position of an activity v in sequence Ω_k or inserting v into sequence Ω_k, Ω is split into two subsets: the sets of activities preceding and succeeding v, respectively. Consequently, a move on v can be specified by two subsets Pre(v) and Suc(v)

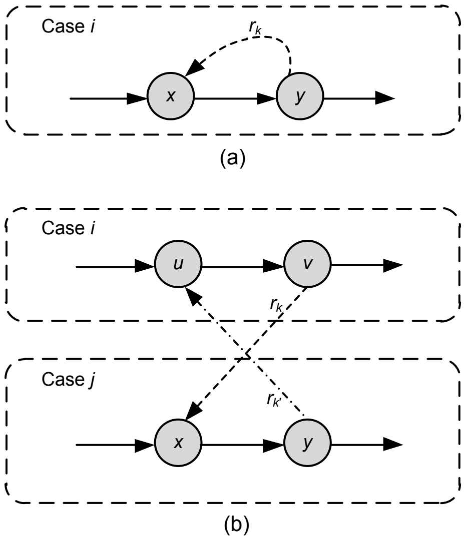

Fig. 3 Two 'deadlock' examples resulting from infeasible moves: (a) deadlock in the activity sequence of a single resource; (b) deadlock among the activity sequences of multiple resources

of Ω: all activities in Pre(v) precede and all activities in Suc(v) succeed v after the move.

Theorem 1 A move on activity v is a feasible move if and only if Pre(v)∩Suc(v)=∅ and x∈Pre(v) implies that y∈Pre(v) ∀y∈Ω with $y \prec x$.

Proof Direction 'only if': Let a move on v be an arbitrary feasible move. Pre(v)∩Suc(v)=∅ must hold because there is no cycle in G. Let us prove that x∈Pre(v) implies that y∈Pre(v) for all y∈Ω with $y \prec x$. Suppose not, and let x and y be a pair of activities that violate this condition. Then there exists a path in G from v to y, for y∈Suc(v). On the other side, $y \prec x$ and x∈Pre(v), and there is a path from v to y and from y to x. Thus, G contains a directed cycle, which is a contradiction.

Direction 'if': Suppose that there is a cycle C∈G after a move on activity v. Because G is an acyclic, it follows that v is on C. We claim that C must contain two activities, x and y, such that x∈Pre(v) and y∈Suc(v). Suppose not, i.e., assume that C does not contain either activities from Pre(v) or activities to Suc(v). Suppose that C∩Pre(v)=∅. There is a case in which C=(..., v, ..., y, ...), for some y∈C∩Suc(v). Thus, there is no path directed from y to v in G. Hence, C is not a cycle with y∈Suc(v), similar to the case C∩Suc(v)=∅. Thus, C must contain two activities x and y such that x∈Pre(v) and y∈Suc(v).

Because x∈Pre(v)∩C and y∈Suc(v)∩C, there must exist a path from x to v and a path from v to y in G. Moreover, x∈Pre(v) and Pre(v)∩Suc(v)=∅ imply x∉Suc(v), and similarly, y∉Pre(v). On the other side, $y \prec x$ must hold if C is a directed cycle. Thus, y∈Pre(v), which is a contradiction.

4.4 Neighbor evaluation

After a feasible move, a neighbor is generated, which should be evaluated by the objective function in problem (1). Note that when changing sequences Ω_k after a move on activity v, arc lengths in G may change. For this reason, it is necessary to recompute the exact values for the start and completion times of each case after each move. In this study, we give Algorithm 1 to evaluate a neighbor after a feasible move based on the objective value.

Algorithm 1 Neighbor evaluation

```
1  Let Ω be the set of arrival activities and initially add the
     first activity of each case to Ω
2  Sort activities of Ω in nondecreasing order according to
     their arrival time
3  If Ω≠∅ then
     Let a_ijk be the first element of Ω
4    Refer to resource r_k to obtain the immediate previous
       activity completion time of a_ijk in its work list, and
       name it t*
5    If a_ijk is the first activity in the work list then
6      Set t*=0
7    End If
8    For each immediate precedent activity a' of a_ijk then
9      Let t_a' be the completion time
10     If a_ijk is the first activity of case i then
11       Set t_a=τ_i and let t*=max(t*, t_a')
12     End If
13   End For
14   Let t* be the start time of a_ijk, and let the sum of t* and
       the processing time of r_k on a_ijk be the completion
       time of a_ijk
15   Add the immediate successor activities of a_ijk of case i
       to Ω
16   If the immediate next activity a' of a_ijk in the work list
       of resource r_k is not in the list of Ω then
17     Add a' to Ω
18   End If
19   Go to step 2
20 End If
21 Calculate the objective value according to the objective
     function defined in problem (1)
```

4.5 Tabu list

The Tabu list is used to prevent the search from cycling between solutions. The Tabu list is based on a 'first-in first-out' (FIFO) queuing rule to store recently searched plans. We use the forbidding strategy to manage the plans entering a Tabu list and avoid cycling and local minima by forbidding certain moves during the most recent computational iterations. We also use an aspiration strategy to enable a plan that has been forbidden by the Tabu list to become acceptable if it satisfies a certain criterion, to provide some flexibility in the forbidding restrictions by leading the search in a desirable direction. Furthermore, our Tabu list has a fixed size. When the list is full, the oldest element in the list is replaced by a new element. We define the stopping criteria for our procedure as follows: if 100 iterations have passed without any improvements in the best solution, then we terminate the TS procedure.

5 Experiments

In this section, we give some simulations. The implementations are coded and compiled in Java and run on a dual quad-core 2.66-GHz Intel Xeon CPU with 1 GB RAM.

5.1 Simulation setting

There is an important issue that needs to be addressed: how to perform numerical experiments based on various aspects of a real process environment. For this issue, we follow the part of the CT scan examination process model described in Section 2. Because the RRSA problem is considered in this study and emergent cases are always started by activity C, we omit the first two activities and assume that the case starts with activity C. Furthermore, we consider only medical equipment and human resources in the simulation because other logical resources such as RIS, PACS, and HIS deal with activities quickly and have insignificant influence on the objective.

At first, we consider the following basic case scenario. A medical practice is operational between 8:00 AM and 12:00 AM. It plans 20 appointment examinations (I_{app}=20) with a probability of missed appointments of ρ=0.1, and the emergent examination arrival rate parameter is λ=1.5/h. Two CT machines CT_1 (16PIE) and CT_2 (64PIE) are qualified to perform activity C, three radiologists r_1, r_2, and r_3 are qualified to conduct activity E, and two radiologists r_3 and r_4 are qualified to conduct activity F. Note that the time of performing each kind of activity is different according to the examination type and resource ability. After discussions with the department managers and analysis of the historical statistical data, we list the time of performing each type of activity by the qualified resources in Table 2. In this simulation, we assume that the probability of occurrence is the same for each examination type in each process case.

5.2 Scheduling policies

We compare our approach with three existing schedule policies, which are listed as follows:

1. FIFO: The FIFO policy attempts to implement an unbiased conflict solver, because it neglects the properties of activities and the state of resources.

2. SPT: The shortest performing time (SPT) policy gives priority to the activity with the shortest imminent performing time. Activities waiting in a queue may cause their dedicated successor resources to be idle. SPT alleviates this risk by reducing the length of the queue in the fastest possible way.

3. SLACK: The slack of a process case is defined as the time span left within its expected duration, assuming that the remaining activities are performed without any delay. Because process cases may wait in front of each resource, the policy 'slack per number of activities remaining' gives priority to the case with the minimum ratio of slack to the number of remaining activities.

5.3 Simulation results and evaluation

First, we simulate the above basic case scenario. The weight for the average flow time α_F is taken as 0.8. The weight for the average resource idle time α_I is taken as 0.1. The average resource idle time has a relatively low weight because of missed examinations. The weight for the average delay α_T is taken as 0.1.

For each policy in each case, we perform 100 simulations, and present the evaluation parameters in Table 3. As shown, the TS-based approach performs better than the three other policies, which confirms the usefulness of the TS-based policy and demonstrates its applicability. For most evaluation parameters, the TS-based approach outperforms the others

Table 2 Average time of performing an activity by the qualified resources obtained after discussions with the department manager and analysis of the historical statistical data

Radiological examination type	Activity*	Time (min)					
		CT_1	CT_2	r_1	r_2	r_3	r_4
Craniocerebral scan	C	20	15	–	–	–	–
	E	–	–	20	20	15	–
	F	–	–	–	–	10	10
Craniocerebral enhanced scan	C	20	15	–	–	–	–
	E	–	–	30	30	20	–
	F	–	–	–	–	15	15
Cervical scan	C	20	15	–	–	–	–
	E	–	–	20	20	15	–
	F	–	–	–	–	10	10
Cervical enhanced scan	C	20	15	–	–	–	–
	E	–	–	30	30	20	–
	F	–	–	–	–	15	15
Chest scan	C	20	15	–	–	–	–
	E	–	–	20	20	15	–
	F	–	–	–	–	10	10
Chest enhanced scan	C	20	15	–	–	–	–
	E	–	–	30	30	20	–
	F	–	–	–	–	15	15
Cardiac scan	C	20	15	–	–	–	–
	E	–	–	30	30	20	–
	F	–	–	–	–	15	15
Cardiac enhanced scan	C	30	20	–	–	–	–
	E	–	–	40	40	30	–
	F	–	–	–	–	20	20
Abdominal scan	C	20	15	–	–	–	–
	E	–	–	20	30	15	–
	F	–	–	–	–	10	10
Abdominal enhanced scan	C	20	15	–	–	–	–
	E	–	–	30	30	20	–
	F	–	–	–	–	15	15
Limb scan	C	20	15	–	–	–	–
	E	–	–	20	20	15	–
	F	–	–	–	–	10	10
Limb enhanced scan	C	20	15	–	–	–	–
	E	–	–	30	30	20	–
	F	–	–	–	–	15	15
Spine scan	C	20	15	–	–	–	–
	E	–	–	20	20	15	–
	F	–	–	–	–	10	10
Spine enhanced scan	C	20	15	–	–	–	–
	E	–	–	30	30	20	–
	F	–	–	–	–	15	15

* Activity C: performing examination; E: transcribing report; F: auditing report. CT_1 and CT_2 are two CT scanners qualified to perform activity C; r_1–r_3 are three radiologists qualified to perform activity E; radiologists r_3 and r_4 are qualified to conduct activity F

Table 3 Comparison of the TS-based approach with the other three policies in the basic case scenario

Method	Average flow time (ms)	Average idle time (ms)	Average tardiness	Average objective value	Standard deviation of objective value	MSE (with 5% confidence interval)
TS	78.53	44.25	0.15	59.42	4.01	58.63 (60.22)
SPT	91.17	39.54	0.97	67.97	5.40	66.90 (69.04)
FIFO	109.81	73.61	39.14	92.06	7.27	90.10 (92.98)
SLACK	113.70	74.96	37.59	94.60	10.06	92.62 (96.61)

Method	Resource utilization (%)					
	CT_1	CT_2	r_1	r_2	r_3	r_4
TS	87.4	81.5	74.23	87.32	81.37	77.55
SPT	88.1	88.2	74.80	80.40	85.60	84.00
FIFO	77.8	98.4	32.20	36.40	88.20	83.00
SLACK	77.3	98.0	31.40	36.10	87.30	82.50

TS: Tabu search based resource allocation method; SPT: shortest performing time policy based resource allocation method; FIFO: first-in first-out policy based resource allocation method; SLACK: slack of a process policy based resource allocation method. MSE: mean squared error of the objective value. CT_1 and CT_2 are two CT scanners qualified to perform activity C; r_1–r_3 are three radiologists qualified to perform activity E; radiologists r_3 and r_4 are qualified to conduct activity F

with a lower mean squared error (MSE) of the objective value. Note that for the average resource idle time and delay, the TS-based approach is close to the SPT policy, and outperforms the FIFO and SLACK policies.

Here we show the advantages of the TS-based approach in solving the RRSA problem in the basic case scenario. To demonstrate that the TS-based approach can help optimize RRSA in more uncertain conditions, we look at what happens if we change some parameters. The changes are made such that the total number of examination cases does not change. The number in the basis case scenario is $I_{\text{app}}(1-\rho)+\lambda T=20\times0.9+1.5\times4=24$. Here, we change two parameters, i.e., λ and ρ, and I_{app} and ρ, at a time respectively. Other parameters are not changed. The results are given in Table 4. As shown, if ρ becomes larger (thus λ increases), the average flow time, idle time, and delay all become larger, because of the higher uncertainty. Furthermore, Table 4 shows that the TS-based approach performs better than the other three policies in conditions with higher uncertainty.

The simulation results indicate that the TS-based approach offers a substantial advantage in optimizing RRSA. Both patient throughput and resource utilization benefit substantially from the TS-based approach. This advantage is especially compelling in scenarios of higher uncertainty, in which the information necessary for performing RRSA efficiently is not available in advance and a good solution can be obtained by TS. Even when this is not the case, as in the basic case scenario, the TS-based approach performs well.

Note that the results presented here are obtained in simulation. In particular, although the simulator strives to capture many of the intricacies that make efficient RRSA a challenge in REP execution, it also glosses over many complicated aspects of real scenarios. For example, in the simulator, the time required to perform an activity is a deterministic metric function, whereas in a real scenario it depends on the state of the availability, preference, and competence of the resources, their cooperation with other resources, etc. In addition, the simulator assumes that the activities of REP never need more than one resource simultaneously, thus avoiding the need to reason about locks. These, and many other issues, need to be addressed before the TS-based approach can be deployed in practice. Nonetheless, simulation results in this study suggest that an effective RRSA solution may play an essential role in improving the performance of REP execution.

6 Related work

Three types of scheduling problems have been studied in the literature related to hospital management: staff scheduling, lab capacity planning, and special medical equipment planning.

Table 4 Outcome value (λ against ρ)

Method	Average flow time (ms)			Average idle time (ms)			Average tardiness (ms)		
	λ=2.0 ρ=0.2	λ=2.5 ρ=0.3	λ=3.0 ρ=0.4	λ=2.0 ρ=0.2	λ=2.5 ρ=0.3	λ=3.0 ρ=0.4	λ=2.0 ρ=0.2	λ=2.5 ρ=0.3	λ=3.0 ρ=0.4
Tabu	76.90	76.77	78.02	39.63	42.27	40.31	0.15	0.26	0.67
FIFO	91.53	91.96	91.70	40.81	43.77	49.63	2.92	1.92	1.17
SPT	107.67	108.39	109.12	72.98	72.85	72.82	34.44	35.68	36.64
SLACK	111.49	112.82	112.17	74.39	74.02	74.00	36.99	36.11	31.00

Method	Average objective value			MSE (with 5% confidence interval)		
	λ=2.0, ρ=0.2	λ=2.5, ρ=0.3	λ=3.0, ρ=0.4	λ=2.0, ρ=0.2	λ=2.5, ρ=0.3	λ=3.0, ρ=0.4
Tabu	59.60	59.99	58.95	58.85 (60.35)	59.20 (60.75)	58.16 (59.75)
FIFO	68.74	68.93	69.38	67.59 (69.89)	67.68 (70.18)	68.08 (70.69)
SPT	89.55	90.30	90.94	87.11 (91.99)	87.56 (93.03)	88.52 (93.35)
SLACK	92.88	93.60	92.12	90.42 (95.34)	91.07 (96.13)	89.90 (94.34)

TS: Tabu search based resource allocation method; SPT: shortest performing time policy based resource allocation method; FIFO: first-in first-out policy based resource allocation method; SLACK: slack of a process policy based resource allocation method. MSE: mean squared error of the objective value

Most studies on the staff scheduling problem are related to hospital labor regulation policies, especially nurse scheduling and physician scheduling (Cheang et al., 2003). In these studies generally linear programming (LP), integer programming (IP), and binary integer programming (BIP) techniques have been adopted to solve the scheduling problem. Heuristic approaches have also been adopted, such as the Tabu search (Glover, 1986), the column generation approach (Bard and Purnomo, 2005), and the genetic algorithm (Aickelin and Dowsland, 2004).

Studies on lab capacity planning and special medical equipment planning problems have focused on the issues related to capacity planning (Marinagi et al., 2000; Roland et al., 2010). They have adopted a variety of techniques in their solution processes: simulation (Kim et al., 2000), rule-based artificial intelligence (AI) (Spyropoulos, 2000), or knowledge-based agents (Marinagi et al., 2000). Decision support systems (DSSs), for which both simple rules (e.g., FIFO, priority, and random) and dynamic approaches to automatic optimization of resource calendars are adopted to deal with emergency situations, are often used to facilitate planning and scheduling of medical resources (Oddi and Cesta, 2000; Vermeulen et al., 2009).

The RRSA problem considered in this study is different from the above scheduling problems in that its objectives and constraints are very different. The purpose of RRSA is to minimize patient examination execution time and shorten resource idle time and delays in medical examination process execution, which are necessary for high patient throughput in the radiology department.

The RRSA problem is very similar to the resource-constrained job shop scheduling problem (Vilcot and Billaut, 2008). A job shop problem considers n jobs which arrive at the shop at certain points in time, referred to as release dates. The problem consists of scheduling the jobs on m machines with respect to technological constraints whilst pursuing a given objective, i.e., flow time-dependent objectives (Vilcot and Billaut, 2008). Checking the feasibility of the hospital examination request for a patient can be comparable to checking the availability of a work center for a customer demand. Like the RRSA problem, a general job shop scheduling problem can be formulated as a BIP or IP model. Very efficient heuristics have been developed for these problems (Vilcot and Billaut, 2008). Many algorithms benefit from a schedule representation known as ‘disjunctive graph formulation’ (Błażewicz et al., 2000). Particularly, local-search algorithms like simulated annealing and Tabu search have been applied with great success.

On the other side, there are differences due to different context. The hospital is a quite unpredictable environment where there are multiple medical resources involved in the execution of REP. To build an efficient solution to the RRSA problem, resource characteristics need to be considered at first, i.e., the capacity of medical equipment, the capability of staff, etc. Moreover, unpredictable events (e.g., emergent or missed examinations) always occur, which is an additional element of difficulty in defining an efficient scheduling and allocation strategy for medical resource usage.

The approach treated uses TS as it is found in scheduling machines in industry or in processes in operating systems, and combines it with results from graph theory. Furthermore, we emphasize that we do not provide hospital patient scheduling (Vermeulen et al., 2009). We assume that the patient schedules already exist, and that we only have to schedule and allocate qualified resources to perform REP execution.

7 Conclusions

In this study, a Tabu search (TS) based approach was proposed to solve the radiological resource scheduling and allocation (RRSA) problem in radiological examination process (REP) execution. Specifically, we presented the details of a CT scan examination process in the Chinese Huzhou Hospital. Minimization of the weighted sum of average examination flow time, resource idle time, and delay is crucial for high patient throughput in the hospital. The scheduling and allocation must be flexible to achieve high service levels for different kinds of scenarios according to resource capacity and ability. We have implemented a realistic system in our case study to analyze the problem and evaluate the approaches. Furthermore, three other scheduling policies were compared with our approach by simulations. Results indicated that our approach outperforms all the benchmarks in terms of solution quality. This merit is more impressive in scenarios with a large number of REP cases and limited resource provision.

The approach proposed here is at an operational level. Study feedback is provided to radiology department managers and we plan to integrate the proposals of this study into the process-aware radiology information system in the Chinese Huzhou Hospital, which will help achieve efficient resource scheduling and allocation in REP execution.

For the moment, our approach considers only emergent or missed examinations and does not consider other uncertainties, i.e., equipment breakdowns or unavailability of human resources. In future research, we hope to add these types of uncertainties to the RRSA problem to activate real-time responses if any of these uncertain conditions arises.

Currently, our approach considers the sequence processes without an alternative process activity structure; i.e., the RRSA problem consists of determining when to execute an activity, and which set of alternative activities to be executed at all (Beck and Fox, 2000). Partial knowledge about the execution of the processes that have an alternative activity structure, makes it difficult to know the actual flow of decision points at the process case instantiation time and typical variance during the execution of individual activities. However, it is desirable to make the best use of the available information to improve resource scheduling and allocation by making the information about future workloads available early, to provide a means of digesting and using this information. From the practical viewpoint, supporting resource scheduling and allocation in the case of all kinds of process structures represents an interesting and desirable direction for future research.

References

Aickelin U, Dowsland KA, 2004. An indirect genetic algorithm for a nurse-scheduling problem. *Comput Oper Res*, 31(5):761-778.
https://doi.org/10.1016/S0305-0548(03)00034-0

Bard JF, Purnomo HW, 2005. Hospital-wide reactive scheduling of nurses with preference considerations. *IIE Trans*, 37(7):589-608.
https://doi.org/10.1080/07408170590948468

Beck JC, Fox MS, 2000. Constraint-directed techniques for scheduling alternative activities. *Artif Intell*, 121(1-2): 211-250.
https://doi.org/10.1016/S0004-3702(00)00035-7

Błażewicz J, Pesch E, Sterna M, 2000. The disjunctive graph machine representation of the job shop scheduling problem. *Eur J Oper Res*, 127(2):317-331.
https://doi.org/10.1016/S0377-2217(99)00486-5

Brah SA, Loo LL, 1999. Heuristics for scheduling in a flow shop with multiple processors. *Eur J Oper Res*, 113(1): 113-122.
https://doi.org/10.1016/S0377-2217(97)00423-2

Cheang B, Li H, Lim A, et al., 2003. Nurse rostering problems—a bibliographic survey. *Eur J Oper Res*, 151(3): 447-460.
https://doi.org/10.1016/S0377-2217(03)00021-3

Duftschmid G, Miksch S, Gall W, 2002. Verification of temporal scheduling constraints in clinical practice guidelines.

Artif Intell Med, 25(2):93-121.
https://doi.org/10.1016/S0933-3657(02)00011-8

Elmaghraby SE, 1993. Resource allocation via dynamic programming in activity networks. *Eur J Oper Res*, 64(2): 199-215. https://doi.org/10.1016/0377-2217(93)90177-O

Glover F, 1986. Future paths for integer programming and links to artificial intelligence. *Comput Oper Res*, 13(5): 533-549. https://doi.org/10.1016/0305-0548(86)90048-1

Huang ZX, van der Aalst WMP, Lu LD, et al., 2010. An adaptive work distribution mechanism based on reinforcement learning. *Exp Syst Appl*, 37(12):7533-7541. https://doi.org/10.1016/j.eswa.2010.04.091

Huang ZX, Lu XD, Duan HL, 2011a. Mining association rules to support resource allocation in business process management. *Exp Syst Appl*, 38(8):9483-9490. https://doi.org/10.1016/j.eswa.2011.01.146

Huang ZX, van der Aalst WMP, Lu XD, et al., 2011b. Reinforcement learning based resource allocation in business process management. *Data Knowl Eng*, 70(1):127-145. https://doi.org/10.1016/j.datak.2010.09.002

Kim SC, Horowitz I, Young KK, et al., 2000. Flexible bed allocation and performance in the intensive care unit. *J Oper Manag*, 18(4):427-443. https://doi.org/10.1016/S0272-6963(00)00027-9

Kubzin MA, Potts CN, Strusevich VA, 2009. Approximation results for flow shop scheduling problems with machine availability constraints. *Comput Oper Res*, 36(2):379-390. https://doi.org/10.1016/j.cor.2007.10.013

Marinagi CC, Spyropoulos CD, Papatheodorou C, et al., 2000. Continual planning and scheduling for managing patient tests in hospital laboratories. *Artif Intell Med*, 20(2): 139-154. https://doi.org/10.1016/S0933-3657(00)00061-0

Mika M, Waligóa G, Węglarz J, 2008. Tabu search for multimode resource-constrained project scheduling with schedule-dependent setup times. *Eur J Oper Res*, 187(3): 1238-1250. https://doi.org/10.1016/j.ejor.2006.06.069

Negenman EG, 2001. Local search algorithms for the multiprocessor flow shop scheduling problem. *Eur J Oper Res*, 128(1):147-158. https://doi.org/10.1016/S0377-2217(99)00354-9

Oddi A, Cesta A, 2000. Toward interactive scheduling systems for managing medical resources. *Artif Intell Med*, 20(2): 113-138. https://doi.org/10.1016/S0933-3657(00)00060-9

Roland B, di Martinelly C, Riane F, et al., 2010. Scheduling an operating theatre under human resource constraints. *Comput Ind Eng*, 58(2):212-220. https://doi.org/10.1016/j.cie.2009.01.005

Salimifard K, Wright M, 2001. Petri net-based modelling of workflow systems: an overview. *Eur J Oper Res*, 134(3): 664-676. https://doi.org/10.1016/S0377-2217(00)00292-7

Spyropoulos CD, 2000. AI planning and scheduling in the medical hospital environment. *Artif Intell Med*, 20(2): 101-111. https://doi.org/10.1016/S0933-3657(00)00059-2

Verhoeven MGA, 1998. Tabu search for resource-constrained scheduling. *Eur J Oper Res*, 106(2-3):266-276. https://doi.org/10.1016/S0377-2217(98)80001-5

Vermeulen IB, Bohte SM, Elkhuizen SG, et al., 2009. Adaptive resource allocation for efficient patient scheduling. *Artif Intell Med*, 46(1):67-80. https://doi.org/10.1016/j.artmed.2008.07.019

Vilcot G, Billaut JC, 2008. A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem. *Eur J Oper Res*, 190(2):398-411. https://doi.org/10.1016/j.ejor.2007.06.039

Zhang JY, Lu XD, Nie HC, et al., 2009. Radiology information system: a workflow-based approach. *Int J Comput Assist Radiol Surg*, 4(5):509-516. https://doi.org/10.1007/s11548-009-0362-6