PATIENT SCHEDULING IN THE PATHOLOGY LABORATORY

Milad Baghersad¹, Amirhossein Meisami², Ali Azadeh¹, Mehdi Hosseinabadi Farahani¹
¹Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
milad_baghersad@ut.ac.ir
²Department of Industrial and Systems Engineering, Texas A&M University, College Station
meisami@neo.tamu.edu

ABSTRACT

This paper addresses a patient scheduling problem in the pathology laboratory. Because of partial precedence constraints of laboratory tests, the problem is formulated as a semi-online hybrid shop scheduling problem and a mixed integer linear programming model is proposed. Also, a genetic algorithm is developed for solving the problem.

Keywords: Laboratory tests; Patients scheduling; Genetic algorithm; Response surface methodology

INTRODUCTION

Laboratory tests are medical services that include testing samples of blood, urine, saliva, and various tissues or substances in the patient’s body. A laboratory test may be taken for one or more of the following reasons: 1) to verify a suspected diagnosis e.g. measuring plasma glucose level to confirm diabetes mellitus, 2) to exclude a diagnosis e.g. a negative urine culture may rule out bladder and kidney bacterial infection, or 3) to check the effectiveness of the treatment or drugs. Planning and scheduling patients requesting for tests is a cumbersome task because it necessitates concerted cooperation of different parts in the laboratory to minimize patients waiting time and maximize laboratory resources utilization (Marinagi, Spyropoulos, Papatheodorou, & Kokkotos, 2000).

Each laboratory may have several types of tests such as hematology, immunology, chemistry and microbiology and patient who arrives to the laboratory may take test of blood, urine, saliva, sperm or stool. Unlike the offline problems, in this situation the arrival times of patients are not known in advance. Nevertheless, when a patient arrives, types of tests that the patient must take are known. The sequence of some of these tests is restricted (e.g., the urine test must precede the stool test) and some tests can be performed without predefined precedence constraints. Thus, problem can be formulated as a semi-online hybrid shop scheduling problem (SHSSP). Figure 1 illustrates the SHSSSP using the example of two patients. Patient 1 is male and requires four samples of tests, where test D cannot be taken before test B, while tests A and F can be taken at any order. Meanwhile, patient 2 is female and requires three samples of tests, where test E cannot be taken before test C, while test A can be taken at any order.
A mixed integer linear programming (MILP) model is developed to formulate the problem and a genetic algorithm is then proposed to solve the formulated semi-online hybrid shop scheduling problem. To estimate the validity of the algorithm, a lower bound is also calculated and the results obtained by the proposed algorithm on randomly generated problems are compared to the lower bound.

The remainder of the paper is organized as follows. Section 2 provides a review of the literature related to patient scheduling. Mathematical formulation is proposed in section 3. Proposed GA and its operators are described in section 4. For verification, a lower bound is calculated in section 5 and finally, Section 6 draws some conclusions from this study.

**LITERATURE REVIEW**

Providing health services to patients in hospitals and laboratories is becoming gradually more vital from the managerial point of view. While reducing costs and improving financial assets are the main goals of hospital managers, maximizing patients’ satisfaction level is also a crucial issue (Cardoen, Demeulemeester, & Beliën, 2010). In order to achieve these goals, hospital resources and infrastructure must be managed efficiently and cost effectively. (Spyropoulos, 2000) and (Chien, Tseng, & Chen, 2008) have classified hospital infrastructure into four following groups. 1) The personnel, including physicians of various specialties, nurses, specific technicians, and other support personnel. 2) Intensive care units, including the complex infrastructure required to support the monitoring and therapy of intensive care patients. 3) Surgical operation units, with dedicated equipment, tools, and procedures for different operations. 4) Laboratories, such as X-ray, ultrasound, magnetic tomography, biochemistry and radiology. Several studies have been devoted to planning and scheduling resources and operations in different parts of hospital. Research in the field of scheduling in hospital can be classified into two main groups: patient scheduling and resource scheduling such as nurses, surgeries or operating rooms scheduling. Since this research focuses on scheduling patients in laboratories, studies on patient scheduling are briefly reviewed.

Reducing waiting time and increasing patient satisfaction are the main objectives of patient scheduling. Since different parts of a hospital encompass different settings and conditions, patient scheduling problems differ from one situation to another. For example, scheduling patients in an emergency room is very different from the other parts partially because of various processes needed by the patients and different priorities of the patients. (Marinagi et al., 2000) have proposed an integrated patient-wise planning and scheduling system for hospital
laboratories examination tests which supports the dynamic and continual nature of the problem. (Chien et al., 2008) have formulated the patient scheduling problem for physical therapy in the rehabilitation department as an off-line hybrid shop scheduling problem. They have developed a GA-based approach embedded with a decision support system for solving the problem. (Chern, Chien, & Chen, 2008) have considered the problem of health examination scheduling to minimize examinee/doctor waiting time and respect time, while taking resource constraints and other limitations such as the sequence and continuity of the examination procedures into consideration. They have designed a heuristic algorithm to solve the health examination scheduling problem efficiently and effectively. (Min & Yih, 2010) have proposed a stochastic dynamic programming model for scheduling patients in a surgical facility with limited capacity considering patients priority. (Petrovic, Morshed, & Petrovic, 2011) have presented a multi-objective optimization for scheduling radiotherapy treatments for categorized cancer patients. They have considered various real life constraints, such as doctors, machine availability, patient’s category, and waiting time targets. The objectives of the proposed model are minimization of average patients waiting times and minimization of average length of gaps of waiting time targets. (Turkcan, Zeng, Muthuraman, & Lawley, 2011) have investigated sequential appointment scheduling with service criteria using unfairness measures to capture the inequity among patients assigned to different slots. Other criteria such as expectation and variance of patient waiting time, queue length, and overtime have also been considered. In a recent work, (Koeleman, Bhulai, & van Meersbergen, 2012) have investigated patient and personnel scheduling policies for care-at-home service facilities and proposed a trunk reservation heuristic to control the system.

Some studies focus on patient scheduling at emergency departments. Scheduling patients in emergency department differs from other departments in that patients have different priorities. (Yeh & Lin, 2007) have proposed a simulation model to cover the complete flow for the patient through the emergency department. A genetic algorithm is applied to find a near-optimal nurse schedule based on minimizing the patients’ queue time. (Vermeulen et al., 2009) have solved the online problem of scheduling patients with urgencies and preferences on hospital resources with limited capacity. In their problem, for most patients a diagnostic test must be performed before the next visit to the physician. (Kırış, Yüzügüllü, Ergün, & Alper Çevik, 2010) have developed a knowledge-based reactive scheduling system for emergency departments, considering patients priorities, arrival times, flow time, and doctors work load, with the objective of determining the patients who have higher priorities initially, and then minimizing their waiting times. They have investigated the efficiency of the proposed algorithm in terms of patients waiting times and doctors work load.

Most studies in health care operations scheduling focus on static problems, and typically do not consider online decision making on real settings. Notably, to the best of our knowledge, it is the first time that an applied approach is proposed for laboratories examination tests considering precedence constraints of tests, constraint on number of sites or operators for taking tests (i.e. multi-machine problem), and semi-online nature of problem. In this paper, focusing on real settings, an effective approach is proposed for increasing service quality in the pathology laboratory by reducing patients waiting time and improving operations efficiency.
MATHEMATICAL FORMULATION

Problem description and assumptions

In the laboratories, any patient arriving to the system may have different types of tests. According to scheduling terminology, each patient is considered as a job, each resource (i.e. operators or sites) as a machine and each test as a station. It is supposed that when a patient arrives to the system, the types of tests required by the patient are known. Some of these tests have precedence constraints, while others are open. The problem is how to schedule each arriving patient, such that the total waiting time of the arrived patient and the patients already in the system is minimized. There is restricted number of operators or sites in each station. If a patient goes to a busy station, he/she stays in the queue and does not hang off. Moreover, it is supposed that each kind of test takes a certain amount of time for each patient.

Mixed Integer Linear Programming Model

Mixed integer linear programming (MILP) models can be used to solve the small-sized SHSSP within a reasonable computational time and provide a better understanding of the combinatorial optimization problem. The semi-online patient scheduling problem is formulated as an MILP model.

Notation

\( j \) index for patients \((j = 1, ..., J)\)
\( s \) index for stations \((s = 1, ..., S)\)
\( i_s \) index for machines in station \( s \) \((i_s = 1, ..., M_s)\)
\( J \) total number of patients in the system including the arriving patient
\( R_j \) arrival time of patient \( j \)
\( C_{js} \) Completion time of patient \( j \) on station \( s \)
\( C_j \) Completion time of patient \( j \)
\( Q_j \) set of pairs of tests for patient \( j \) representing predefined sequence \((j = 1, ..., J-1)\)
\( U_j \) set of pairs of tests for patient \( j \) representing precedence constraints
\( X_{jis} \) 1 if patient \( j \) is assigned to machine \( i \) at station \( s \) and 0 otherwise
\( P_s \) processing time at station \( s \)
\( S_j \) set of stations that patient \( j \) must visit
\( Z_{jks} \) 1 if patient \( j \) precedes patient \( k \) in station \( s \) and 0 otherwise
\( Y_{js_1s_2} \) 1 if patient \( j \) visits station \( s_1 \) prior to station \( s_2 \) \((s_1, s_2 \in S_j)\)
\( M \) an arbitrary large constant \((M \to \infty)\).

Formulation

Minimize \( \sum_{j=1}^{J} C_j \) \hspace{1cm} (1)

\( C_{js} - P_s \geq R_j \) \hspace{1cm} \forall s \in S_j \hspace{1cm} (2)

\( C_{js_1} \leq C_{js_2} - P_{s_2} \) \hspace{1cm} \forall s_1, s_2 \in S_j, (s_1 \to s_2) \in Q_j, j = 1, ..., J - 1 \hspace{1cm} (3)
The objective function (1) minimizes total completion time of patients in the system including the arriving patient as well as the patients already in the system. Constraint (2) ensures that start time of the just arrived patient process is greater than or equal to its arrival time. Constraint (3) respects previous patients’ tests sequence that is predefined before the present patient arrival. Constraint (4) specifies that each test is assigned to exactly one machine in a station to complete its process. Constraint (5) calculates the completion time of patients. Constraint (6) respects the partial precedence constraints on the set of tests. Constraint (7) ensures that only one test of the just arrived patient is taken at a time. Constraint (8) guarantees that at most one patient is served by each station at any given time.

METHOD

When the first patient arrives to the system, since there is no other patient in the system, order of patient’s tests does not affect the completion time of the patient and therefore, can be scheduled randomly. By the time the next patient arrives, the tests of the just arrived patient are scheduled according to the order of tests and current situation of the patient already in the system with the objective of minimizing total completion time of all patients in the system. This procedure is repeated once another patient arrives to the system. If the number of tests that patients have to take is low, objective function can be calculated for all possible orders of tests to find the best sequence of tests. However, when number of tests grows, number of possible solutions grows exponentially and the problem cannot be solved by enumeration methods in a reasonable amount of time.

In this study, a genetic algorithm is proposed for solving SHSSP. GAs are probabilistic search techniques inspired by Darwin’s theory of evolution by natural selection. In GAs, while a population of candidate solutions iteratively evolves through generations by the use of genetic operations (i.e., selection, crossover and mutation), some individuals adapt better to the environment and have more possibilities of survival. The general structure of the genetic algorithms is shown in Figure 2.
**Chromosome Representation**

One of the most important steps in designing a GA for a particular problem is to devise a suitable representation scheme showing the solution characteristics. In the proposed algorithm, a solution is represented by an ordered sequence of tests which is a permutation of 1 to total number of tests. However, according to the partial precedence constraints intrinsic to SHSSP, the obtained sequence may be infeasible. Hence, it is necessary to check the feasibility of each chromosome whenever it undergoes any operator. In the case of infeasibility, the tests that have been ordered in a way that violate the precedence constraint are swapped. For example, suppose that a patient has five types of tests and test 2 must precede test 5. Initial sequence of tests is illustrated in Figure 3 (a) in which test 5 is taken prior to test 2. In order to repair this infeasible chromosome, positions of tests 2 and 5 are exchanged. The resulted feasible chromosome will be as Figure 3 (b).

![Figure 3](image-url)

**Figure 3** An exemplary chromosome (a) before feasibility check, and (b) after feasibility check.
**Genetic operators**

**Crossover**

Crossover is the main genetic search operator exploiting the information embedded in selected parents to create offspring. The most important issue in designing a crossover operator is that it can preserve the part of the parents’ genetic code that is responsible for their high fitness. The crossover operator used in the proposed algorithm is the linear order crossover (Falkenauer & Bouffouix; 1991). The linear order crossover works as follows:

**Step 1** Select a subsequence of tests from parent 1 randomly and copy it into the corresponding positions of the offspring.

**Step 2** Remove the tests from parent 2 which are held in common with the selected subsequence.

**Step 3** Copy the tests of parent 2 orderly into the unfixed positions of the offspring.

The second offspring is generated by exchanging the roles of the parents. An example of the crossover is illustrated in Figure 4.

![Figure 4 Crossover](image)

**Mutation**

Mutation operator produces random changes in a chromosome in order to keep the diversity of the population in a reasonable level. In the proposed algorithm, the swap mutation is used in which two genes are randomly selected and their values are exchanged. An example of swap mutation is shown in Figure 5.

![Figure 5 Swap mutation](image)
**Selection**

The main feature of a good selection mechanism is that an individual with higher fitness would be selected with higher probability. Roulette wheel approach (Reeves, 1995) is adopted as the selection mechanism for the proposed algorithm.

**Decoding Procedure**

In order to calculate the objective function, chromosomes are decoded into actual schedules. The decoding procedure is designed specifically to calculate the total completion time of all patients in the system. The fitness value calculation is described as follows:

*Step 1*  
Set the just arrived patient as the current patient.

*Step 2*  
Calculate finish time of all tests being currently taken by the patients in the system according to their situation.

*Step 2*  
Based on the number of sites or operators, determine start time of the current patient according to the sequence of tests. Calculate the completion time of the current patient on the current station.

*Step 3*  
Set the patient with minimum completion time as the current patient.

*Step 4*  
1) If the current patient does not have any other test, save the completion time and remove the patient from system, else go to Step 2.  
2) If there is not any other patient in the system go to Step 5, else go back to Step 3.

*Step 5*  
Calculate sum of completion time of all patients, and stop.

**Verification**

In order to verify the proposed approach, the algorithm has been run on 10 randomly generated test problems. The data related to the test problems are shown in Table 3. In Table 3, the problem size shows the number of operations multiplied by the number of patients in the system. Number of machines at each station is considered to be 1 or 2 at random and processing time of operations are random integer numbers between 150 and 800. The algorithm has been run 10 replicates for each test problem, and the best and the average solutions can be seen in Table 3.
### Table 1: Comparison between lower bound and the proposed algorithm

<table>
<thead>
<tr>
<th>No.</th>
<th>Problem Size</th>
<th>Lower bound</th>
<th>Proposed algorithm</th>
<th>Gap %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Best solution</td>
<td>Average</td>
</tr>
<tr>
<td>1</td>
<td>8×15</td>
<td>19580</td>
<td>20575</td>
<td>20575</td>
</tr>
<tr>
<td>2</td>
<td>9×35</td>
<td>156517</td>
<td>160402</td>
<td>160578</td>
</tr>
<tr>
<td>3</td>
<td>10×20</td>
<td>100206</td>
<td>103990</td>
<td>104180</td>
</tr>
<tr>
<td>4</td>
<td>11×33</td>
<td>91176</td>
<td>94801</td>
<td>94821</td>
</tr>
<tr>
<td>5</td>
<td>12×30</td>
<td>134730</td>
<td>141294</td>
<td>141610</td>
</tr>
<tr>
<td>6</td>
<td>12×50</td>
<td>347273</td>
<td>352112</td>
<td>352975</td>
</tr>
<tr>
<td>7</td>
<td>14×25</td>
<td>117088</td>
<td>120599</td>
<td>120810</td>
</tr>
<tr>
<td>8</td>
<td>16×30</td>
<td>128472</td>
<td>133134</td>
<td>133280</td>
</tr>
<tr>
<td>9</td>
<td>17×33</td>
<td>171158</td>
<td>171558</td>
<td>176700</td>
</tr>
<tr>
<td>10</td>
<td>21×50</td>
<td>432320</td>
<td>445386</td>
<td>445600</td>
</tr>
</tbody>
</table>

Average gap: 3.4%

In order to examine the efficiency of the proposed algorithm, the obtained results have been compared to a lower bound on the optimal solution. To calculate the lower bound, total completion time of jobs already in the system is aggregated with the sum of total processing times and release time of the just arrived job. In fact, the lower bound is calculated by releasing machines constrains for the just arrived patient as follow:

\[
LB = \sum_{j=1}^{J-1} C_j + R_j + \sum_{s=1}^{S} p_{js} \tag{9}
\]

where \( p_{js} \) is the processing time of the just arrived patient on station \( s \).

**Theorem 1** Equation (9) is a lower bound for the total completion time of jobs in semi-online hybrid shop scheduling problem.

**Proof** Total completion time of jobs can be considered as the sum of total completion time of jobs (patients) already in the system and the completion time of the just arrived job as follow:

\[
\sum_{j=1}^{J} C_j = \sum_{j=1}^{J-1} C_j + C_j \tag{10}
\]

Furthermore, it is evident that the completion time of the just arrived job cannot be less than the sum of its total processing times and its release time. Hence,

\[
C_j \geq R_j + \sum_{s=1}^{S} p_{js} \tag{11}
\]

According to equations (10) and (11), it can be easily concluded that
\[
\sum_{j=1}^{J} C_j \geq \sum_{j=1}^{J-1} C_j + R_j + \sum_{i=1}^{S} p_{ji} \quad (12)
\]

The above equation implies that total completion time of jobs already in the system aggregated with the sum of total processing times and release time of the just arrived job is a lower bound for the completion time of jobs.

Table 3 shows the comparison of the results obtained by the proposed algorithm and the lower bound. Solution gap is measured by the mean percentage difference from the lower bound. The average gap of the solutions to the lower bound is 3.4%. Results show that the gap depends on the time required for taking tests, i.e. the gap grows as the total processing times increases. In the lower bound calculation, the total completion time of the patients already in the system is exactly calculated and the completion time of the just arrived patient is estimated by the total time required for tests disregarding machines constrains. However, in real conditions the patient must wait in each stage until a machine or operator becomes idle. When the processing time increases, the waiting time of the just arrived patient becomes longer and therefore, the gap increases.

**Parameters Setting**

GA performance strongly relies on the parameters setting. Developing successful methods for setting parameters of GA is currently one of the most attractive research fields in computational intelligence. As mentioned by (Schaffer, Caruana, Eshelman, & Das, 1989), the optimal parameters setting vary from problem to problem. Experimental design approaches such as full factorial design ((Gupta, Gupta, & Kumar, 1993) and Taguchi method (Sun, 2009) (Hsieh, Chou, & Wu, 2001) have been successfully applied to find the optimal operating parameters in the GA. In this study, RSM, one of the most popular optimization methods, is used for parameters optimization in GA. RSM is a useful experimental approach which combines statistical inference, mathematical techniques and experimental strategies for optimizing a response of interest influenced by several variables (Baş & Boyacı, 2007). RSM is a powerful tool for optimizing individual factors and determining the optimum response value. It provides an understanding of how the independent variables affect the response, determines possible interrelationship among the independent variables, and generates a mathematical model which describes the processes (Farooq Anjum, Tasadduq, & Al-Sultan, 1997); (Myers & Montgomery, 1995). In this study, RSM is implemented in three following steps:

**Step 1** Determine independent parameters and their preliminary levels.

**Step 2** Select the appropriate experimental design and predict the model equation as follows:

2.1 Run the proposed algorithm around center point and set a first-order model. If curvature test is significant, set a second-order model and go to Step 3.

2.2 Move sequentially along the direction of maximum decrease in response. When the response starts to get worse, stop and set the last point as center, go to (2.1).

**Step 3** Determine the optimum point and check it.
In Step 1, after a preliminary analysis of the GA algorithm, the three most commonly studied GA parameters, including population size, crossover rate and mutation rate are considered as design factors and the number of iterations is selected enough large regarding the convergence behavior of the algorithm. RSM is applied on various problems with different sizes to determine the optimum parameters of GA. For example, suppose a problem with 26 patients (i.e. there are 25 patients already in the system when the new patient arrives to the system) and 15 different tests. Engineering judgment is used and after preliminary analysis, the initial levels of parameters are selected. Table 1 shows the initial levels of GA parameters when the number of iterations is 60.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>[30,50]</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>[0.60,0.70]</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>[0.15,0.25]</td>
</tr>
</tbody>
</table>

In step 2, the proposed algorithm is run for 10 replicates in each point. A $2^3$ factorial design augmented by five center points is adopted to determine the first-order model. In this example, the curvature test is not significant. Therefore, the new center point is identified by moving sequentially along the direction of maximum decrease in the response by a predetermined step size. The step size is determined by experimenter based on the problem specific knowledge or other practical consideration (Montgomery, 2001). When the response starts to get worse, the last point is selected as center. This procedure is repeated until curvature test is significant. Then, a second-order model is set and in step 3, the optimal point is found. All experiments are executed by Design-Expert 7.0 software. Table 2 shows the optimum levels of GA parameters based on the obtained results for different problems.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>57</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.69</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.29</td>
</tr>
</tbody>
</table>

CONCLUSION

Focusing on real settings, a patient scheduling problem in the pathology laboratory has been formulated as a semi-online hybrid shop scheduling problem, and a mixed integer linear programming model has been proposed. The objective is to minimize total completion time of patients in the system. A genetic algorithm (GA) has been developed for solving the problem and response surface methodology has been used for optimizing GA parameters. A lower bound on the objective value has been calculated for the problem and several experiments have been conducted to verify the validity and to evaluate the efficiency of the proposed algorithm. Further research can be done on developing a decision support system for semi-online patient scheduling problem in the pathology laboratory.
REFERENCES


