A REACTIVE SCHEDULING FOR INTESIVE CARE UNITS

[(a)] **Erhan Kozan**

[(a)] School of Mathematical Sciences, Queensland University of Technology, Australia

[(a)] e.kozan@qut.edu.au

ABSTRACT
This paper leads to significant improvement in intensive care units (ICU) operating efficiency and productivity by optimising scarce resources. Scheduling of patients in the ICU is complicated by the two general types; elective surgery and emergency arrivals. The job shop approach to reactive scheduling system for ICU promises considerable benefits over existing approaches, and allows problems of large size and complexity to be solved with great accuracy. A general parallel machine job shop model is proposed that fits uncertain arrivals with priorities and the allowance of rejection events at Intensive Care Units. An integer programming model is proposed to handle uncertain arrivals with priorities and while allowing rejections.

Keywords: Scheduling, Health Services, Integer Programming

1. INTRODUCTION

There are a number of studies that have focussed on the ICU. Operations research issues investigated include scheduling of patients and resources, allocation of limited resources and physical design of the facility. These will be discussed as well as general scheduling and resource allocation methods that have been applied to hospitals.

Kim and Horowitz (1999, 2000 and 2002) describe models which are the closest to our proposed model. However they do not encompass the whole of the operating theatre (OT) and ICU. In these studies the ICU is modelled completely, and patients who require the ICU after a surgical intervention are included. Their focus is also on balancing deterministic and stochastic arrivals. To accomplish this they have used a quota system to specify the number of beds available to deterministic arrivals each day. They experimented with one or two weeks scheduling windows. Simulation was used to compare the different scenarios. For the particular hospital studied it was found that a scheme allocating two beds on Monday and Friday, and one bed on Tuesday, Wednesday and Thursday with bookings taken for beds over a two week period was optimal. The proposed model will be looking at scheduling patients on a day by day basis, whereas this model determines the best resource allocation for the ICU.

Similar to research undertaken by Kim and Horowitz (2002) and Ridge et al. (1998) develops a simulation model of the ICU focussing on minimising the number of deterministic arrivals that are rescheduled. Notable inclusions in the model are that rescheduled surgeries re-enter the model, and a queuing theory model is used to verify the output. Some sensitivity analysis was performed to ascertain the effects that important variables had on the system. These variables were the number of beds in the ICU, length of reschedule times and the number of beds reserved for emergency admissions. The results were intended to be used as part of a decision analysis tool to decide allocation of beds. While the main focus was on the number of emergency patient transfers it was concluded a more effective patient admission scheduling system could benefit the hospital being analysed. Due to the stochastic nature of arrivals to the critical care facility it is a requirement to reduce their effect on the objective of the schedule.

Sahinidis (2004) reviews the theory and methods developed to cope with the complexity of optimisation problems under uncertainty. The main approaches to handle uncertainty are stochastic programming, robust stochastic programming, fuzzy programming and stochastic dynamic programming.

Stochastic programming uses a two-stage solution system, where the decision variables are partitioned into two sets. The first stage variables are those that can be decided before any uncertainty is realised. Once the random events impact the system improvements can be made by selecting second stage or recourse variables at some cost. The objective is to minimize the sum of the first stage costs and the expected second stage costs. For problems with continuous parameter distributions convexity properties of the recourse function have been used by Infanger (1994); and Shapiro and Homem-de-Mello (1998) to develop sampling based decompositions and approximation schemes.

Sand and Engell (2004) use a two-stage stochastic integer programming model on a moving horizon to schedule a flexible chemical batch process. They found that most previous work used very defensive strategies to generate robust off-line scheduling which require little emphasis on adjustment in real-time. They look at recourse actions as further opportunity for optimisation.

The system is affected by four types of uncertain processing times; deviation in product quality; machine breakdowns; product demand changes. A tree of possible future schedule horizons was computed to find the best first horizon schedule.

Engels and Karger (2003) look at the objective of minimizing the sum of the weighted completions times of jobs scheduled plus the sum of jobs rejected. Both these papers use a dummy machine to place all those jobs that are not scheduled. This machine is not constrained in the same way as the normal machines to allow any job to be placed on it.

There are numerous factors and systems that influence patient flow (eg: number and rate of patient arrivals, number of beds, length of stay, staffing arrangements, etc.). Patient arrivals fall under two groups, elective and emergency. In the case of hospital the number of patients from the elective and emergency groups is approximately the same. Elective patients are known few weeks in advance. On top of this we have batches of these patients arriving at regular times on weekdays. Emergency patients on the other hand give little or no warning of their need to use the ICU. This uncertainty adds complexity to the scheduling of patients.

The aims of this scheduling model are to: normalise utilisation of the ICU at an increased level; reduce rejection and reschedule rates of patients; and allow real-time adjustment for stochastic arrivals.

Another aspect of the model that adds complexity and sets it apart from many conventional scheduling problems is that some patients may be denied service by the ICU because of insufficient resources.

2. THE MODEL

The model describes the deterministic scheduling of one time window within the system. In the case of the ICU there will be an infinite number of these time windows as the unit never stops processing patients. While the schedule created may be satisfactory for some period of time there will always be unexpected events to manage. To manage these unexpected events we need to create a reactive system made up of many deterministic scheduling steps to keep the ICU running optimally.

Patients are divided into fixed and flexible patients. Fixed patients are those patients currently being treated in the ICU or patients that are in the previous schedule and soon to arrive or fixed for other reasons. Admission time of fixed patients can not be changed. Flexible patients are all other patients, either currently in the previous schedule or newly arriving and their admission time may change.

An integer programming for generating a schedule is developed. The objective is to maximise the utilisation of the ICU and minimising the number of patients rejected. The size of the model is determined by the number of patients, beds and time intervals.

2.1 Notations

i : patients, $i \in \{1 \ldots I\}$

j : beds, $j \in \{1 \ldots J\}$

t : time, $t \in \{0,1,2,\ldots,T\}$

f : number of flexible patients in the schedule

A_j : first available time of bed j

a_i^o : original admission time, $i > f$

a_i : real admission time of patient i

c_i : priority index of patient i

d_i : discharge time of patient i ($d_i = a_i + p_i \ \forall i$)

$$D_j = \begin{cases} 1 & \text{if bed } j \text{ is available on weekends,} \\ 0 & \forall j \end{cases}$$

$$e_i = \begin{cases} 1 & \text{if patient } i \text{ is an elective patient} \\ 0 & \text{otherwise} \end{cases}$$

E : maximum number of elective surgery patients schedule at any particular time

K : total idle time of the system

p_i : length of stay of patient i

r_i : arrival time of patient i

s_j : setup time of bed j

$$x_{ijt} = \begin{cases} 1 & \text{if patient } i \text{ is admitted to bed } j \text{ at time } t, \ \forall i,j,t \ ; i \le f \\ 0 & \forall i,j,t \ ; t \ne a^o_i ; \ i > f \end{cases}$$

w_i : waiting time of patient i

w_i^M : maximum waiting time of patient i

2.2 Objective Functions

The objective is to minimise the number of rejected patients by fulfilling the objective to maximise the utilisation of the unit. By including the waiting time in the objective function (1) and scaling by their priority category we aim to minimise rejection of the higher priority patients. Objective function balances the waiting time against the length of stay of the patient, in effect rejecting the patient if the scaled waiting time becomes too large when compared to the length of stay. The following objective function is derived from the waiting time, priority and idle time functions.

$$\mathit{Minimise} \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=0}^{T} x_{ijt} c_i \left(p_i + w_i \right) \tag{1}$$

2.3 Constraints

Maximum one patient is scheduled in bed j at time t. The inclusion of s_j allows for bed dependent setup times. When programming these constraints for a given t only include jobs that satisfy $t - p_i \geq 0$.

$$\sum_{i=1}^{I} \sum_{s_j=\max(t-p_i,0)}^{t-1} x_{ijs_j} \leq 1 \;\; \forall j,t \tag{2}$$

Equation 3 and 4 ensure that each patient is scheduled once. If a patient is not scheduled Equation 3 is equal to 0, which means the patient is rejected.

$$\sum_{j=1}^{J} \sum_{t=0}^{T} x_{ijt} \leq 0 \qquad for \quad i = 1,\dots,f \tag{3}$$

Equation 4 ensures that each patient is scheduled once. For patients that are fixed in the schedule constraint 4 is used

$$\sum_{j=1}^{J} x_{ija_i^o} = 1 \qquad i = f+1, f+2, \dots, I \tag{4}$$

This constraint combined with setting all the other x_{ijt} values for these patients to 0 restricts any flexibility to just bed choice at their admission time. The admission time is calculated as follows. This will be 0 when the patient is not scheduled.

$$a_i = \sum_{j=1}^{J} \sum_{t=0}^{T} x_{ijt} t \quad for \ \forall i \tag{5}$$

Waiting time is the difference between the arrival time of patient i and their admission time. This value needs to be 0 when the patient is not scheduled. It is also required to limit the maximum waiting time of a patient to a fraction of their length of stay.

$$0 \leq w_i = \sum_{j=1}^{J} \sum_{t=0}^{T} x_{ijt} \left(t - r_i\right) \leq w_i^M \tag{6}$$

Equation 7 ensures that no patient occupies bed j before it becomes available. This constraint is irrelevant for the first time window that is scheduled, but is necessary for subsequent time windows when patients overlap.

$$\sum_{t=0}^{A_j-1} \sum_{i=1}^{I} x_{ijt} = 0 \;\; \forall j \tag{7}$$

Equation 8 ensures that there are no more than the specified amount of elective surgery patients scheduled at one time.

$$\sum_{i=1}^{I} \sum_{j=0}^{J} x_{ijt} e_i \leq E \;\; \forall t \tag{8}$$

Reducing the number of beds in use over the weekends is handled by setting the value of A_j for the beds that should not be used to either midnight of Sunday if that occurs in the scheduling window or to $T+1$ if it does not. This makes sure that the scheduling algorithm will not schedule any patients on these specific beds.

3. IMPLEMENTATION

The model above describes the deterministic scheduling of one time window within the system. In the case of the ICU there will be an infinite number of these time windows as the unit never stops processing patients. While the schedule created may be satisfactory for some period of time there will always be unexpected events to manage. To manage these unexpected events we need to create a reactive system made up of many deterministic scheduling steps to keep the ICU running optimally.

This model was solved with the aid of the CPLEX libraries. Using the input file the software generates a list of patients that will arrive in the time window specified. Patient arrival times and length of stay values are generated and initially don't coincide with the time slots.

The software rounds the arrival times back and the length of stay values out to the closest time. When a list of patients is generated, it is used for differing number of beds to provide a sensitivity analysis to determine how much impact this factor has on the schedule. This can be measured in number of patients rejected, utilisation of the unit.

At time 0 there are no patients in the system so a warm-up period is used to bring the system to a steady state and from that point descriptive statistics can be taken from the schedule that is developed. A schedule is created with elective patients over the scheduling horizon. Subsequent schedules are generated when emergency patients arrive; the warm up period is over; and no emergency patients have arrived for the number of intervals specified by schedule length.

Patients are split into groups at the point of the expected arrival of the new patient. Basically we have those patients that have their admission time and/or bed fixed and those that are flexible. The patients that are currently being treated in the ICU have their bed and admission time fixed. Patients that have not arrived yet may have their admission time fixed by the scheduler if their admission time has been changed a certain number of times, or if they are set to arrive within a certain amount of time. Flexible patients may have their admission time and bed changed or they are rejected from the system.

An initial schedule of fixed and flexible Patients is shown in Figure 1. This information is then used to determine bed availability times for subsequent schedules. Once the new schedule is found this information is fed into the next one and so on. The approach for this re-scheduling step is to generate a new schedule that is as close to the previous one as possible. Fixing patients to reduce the impact of future schedule changes on them is one part of this.

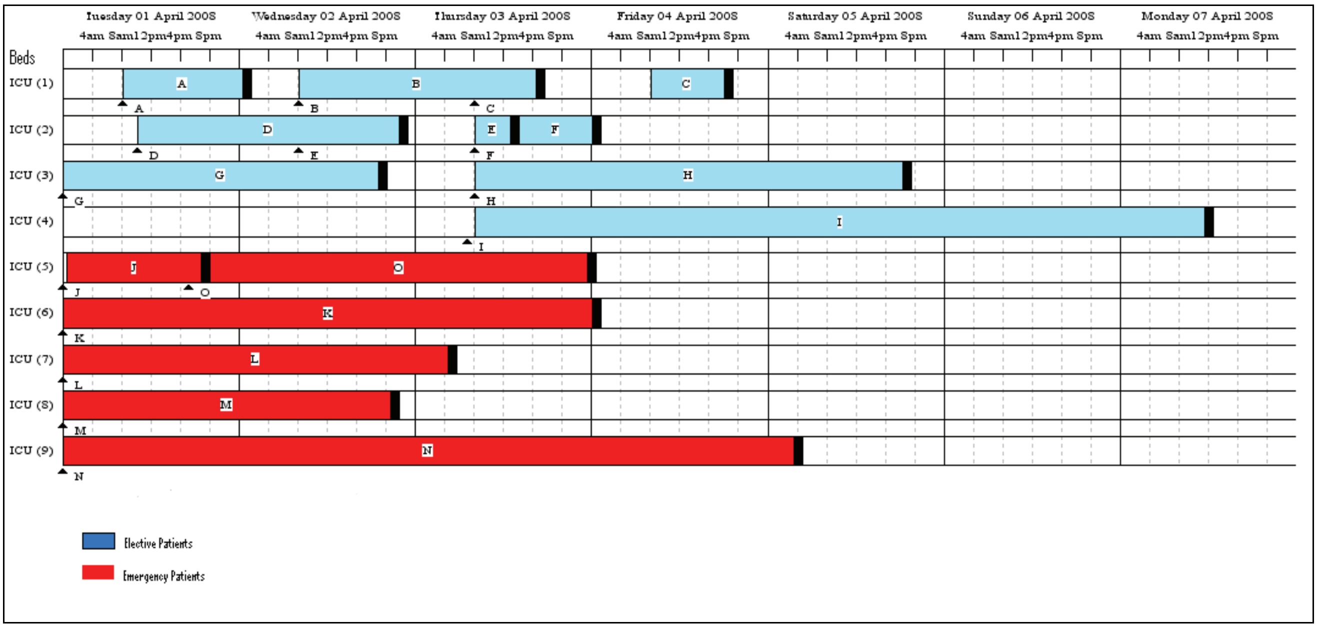

Figure 1: An initial schedule of fixed and flexible Patients

4. CONCLUSIONS

It was our aim to keep the model as general as possible to increase its applicability to other ICU's. The focus of the model is scheduling of humans or jobs that incur a penalty for job start time changes, while allowing for job rejections in a dynamic environment. This model could be modified to fit other units within the hospital, as well as external practices that handle patients in a similar manner.

The next step for this model is to determine the effects of the parameters in the objective function. We can investigate the balance between waiting time and length of stay time has on patient rejections to provide more information to the implementers of the system. The use of the scaling factor on the waiting time value may prove crucial in finding schedules that are good for the system they are applied to.

More investigation should to be done with the schedule window length in relation to patient input parameters and their effect on the objectives of the system. The results above show that there may be a link, and therefore a different schedule length may need to be applied to different hospitals using the system.

The focus of the model is scheduling of humans or jobs that incur a penalty for job start time changes, while allowing for job rejections in a dynamic environment. This model could be modified to fit other units within the hospital, as well as external practices that handle patients in a similar manner.

REFERENCES

Engels, D. W. and Karger, D. R. (2003). Techniques for scheduling with rejection. *Journal of Algorithms,* 49, 175-191.

Infanger, G. (1994). *Planning under uncertainty: Solving large scale stochastic linear program*s. Boyd and Fraser Publishing Co.

Kim, S. and Horowitz, I. (2002). Scheduling hospital services: the efficacy of elective-surgery quotas. *Omega,* 30, 335-346.

Kim, S. and Horowitz, I. (1999). Analysis of capacity management of the intensive care unit in a hospital.

European Journal of Operational Research, 115, 36-46.
Kim, S. and Horowitz, I.(2000). Flexible bed allocation and performance in the intensive care unit. Journal of Operations Management, 18, 427-443.
McHardy, S. and Kozan, E. (2004). Modelling of Intensive Care Units and Operating Theatres in Public Hospitals. *The Fifth Asia-Pacific Industrial Engineering and Management Systems Conference,* Australia.
Ridge, J. C. and Jones, S. K. (1998). Capacity planning for intensive care units. *European Journal of Operational Research*. 105, 346-355.
Sahinidis, N. V. (2004). Optimization under uncertainty: state-of-the-art and opportunities. *Computers and Chemical Engineering,* 28, 971-983.
Sand, G. and Engell, S. (2004). Modeling and solving real-time scheduling problems by stochastic integer programming. *Computers and Chemical Engineering*, 28, 1087-1103.
Shapiro, A. and Homem-de-Mello, T. (1998). A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming,* 81, 301-325.

AUTHORS BIOGRAPHY

Erhan Kozan is a Chair Professor of Operations Research, in the School of Mathematical Sciences at Queensland University of Technology, Australia. He has had 33 years industrial, managerial, teaching and research experience in the areas of Operations Research. He worked with the World Bank Group and the United Nations Development Program. He is the National President for the Australian Society for Operations Research. Professor Kozan has acted as principal investigator for over 22 long-term industrial projects, and 16 competitive national and international research grants in the area of health, finance, production, railways and seaports transportation. He is the author of a book, nine softwares and over 150 articles. He is the editor and associate editor of seven journals. He has supervised over 30 postgraduate research students. He is currently supervising five PhD students in the health and transportation area.