Control and Cybernetics

vol. **43** (2014) No. 4

Time buffers in the open shop problem*†

by

Dorota Kuchta

Wroclaw University of Technology
Institute of Organisation and Management
Wybrzeże Wyspiańskiego 27, 50-370 Wroclaw, Poland
dorota.kuchta@pwr.edu.pl

Abstract: A stability concept for the open shop schedule is proposed. The underlying aim is to protect an organisation against serious problems caused by accumulation of delays in various processes realised in the organisation at the same time. The stability criterion is coupled with that of schedule makespan minimisation. Buffers are used as a tool of ensuring schedule stability. A fuzzy model determining the optimal schedule with respect to both criteria is formulated based on expert opinions as to the desired buffer size and the desired planned makespan. The model is expressed as a mixed integer linear programming model. The approach is illustrated by means of an example.

Keywords: open shop, stable schedule, schedule buffers

1. Introduction

Machine scheduling, including the open shop problem, has been studied for many years in various versions (see, e.g., Baker and Trietsch, 2009; Dong et al., 2013; Panwalkar and Koulamas, 2014). In many cases deterministic models are studied, but "practitioners often view the ignorance of uncertainty and the dynamic elements of the scheduling process as the major source of the gap between scheduling theory and practice", according to Sabuncuoglu and Goren (2009). And they are right to expect the uncertainty and dynamism of the scheduling context to be taken into account in the models proposed in the literature, as there is a long list of phenomena that may and do influence schedules and their realisation.

Sabuncuoglu and Goren (2009) present a list of possible schedule disruption causes: unexpected arrivals, machine breakdowns, processing time variability, due time changes, job cancelation, ready time changes, scraps and waste. All

*The original version of this paper was presented at the BOS 2014 Conference (Operations and Systems Research) in September 2014 in Warsaw.

†Submitted: September 2014; Accepted: November 2014.

these phenomena and many others may cause a deterministic model to be of little help in actual implementation.

Scheduling theory has been trying to incorporate this aspect. This has been done by modelling processing times as random variables or fuzzy numbers (see, e.g., Baker and Trietsch, 2009; Kasperski, Kurpisz and Zieliński, 2012), through simulation of the randomness of such parameters as jobs arrivals, machine disruptions, etc. (for a review see Sabuncuoglu and Goren, 2009). In this context, such schedule features as schedule robustness/stability/sensitivity/security have been introduced and studied (in, for instance, Xiong, Xing and Chen, 2013; Aloulou and Artigues, 2010; Baker and Trietsch, 2009; Herroelen and Leus, 2004; Jensen, 2001). The issue is considered very important and the robustness, security and stability degree of a schedule have recently become important performance measures of schedule goodness, along with the traditional ones, i.e. makespan, flowtime, earliness or tardiness (Sabuncuoglu and Goren, 2009).

Here we concentrate on the concept of schedule stability (robustness is discussed, for instance, in Jensen, 2001; security, in Baker and Trietsch, 2009; flexibility in Aloulou and Artigues, 2010; and all of these aspects are discussed and compared with each other in Sabuncuoglu and Goren, 2009). The concept of stability is concerned with the difference between the initial and the actually realized schedule, see Sabuncuoglu and Goren (2009): “A schedule whose realisation does not deviate much from the initial schedule is called stable”.

The impact of a change in schedule may be serious not only for the jobs included in the very schedule and their immediate users/owners who wait for them being ready, but also to the whole organisation. A single flow/job/open shop schedule is never the only process to be realised in the organisation. There are other schedules to be realised, too, and we have also supporting processes, such as deliveries, and the like. So, if a job in the schedule is processed in another time period than it was expected, it may pose a serious problem for several reasons – the ones linked to the job or the process in question itself (the job or the process may be finished later than expected and the customer may be dissatisfied) and the ones linked to the environment of the whole process. Here we are interested rather in the latter ones. The processing of each job on each machine may require the usage of resources that are also needed in other processes (we mean here, above all, human resources, machine operators for example) in the given organisation. If a job on a machine is late with respect to the initial schedule (even if it is not the last machine in the process considered and eventually the whole process or even the job in question will still be on time), it may cause quite a nuisance with respect to the coordination of all the processes in the organisation, as they may have to wait until the needed resources are free. In other words, it may increase organisation nervousness (Sabuncuoglu and Goren, 2009). Like it is done in multi-project management (see, e.g., Correia and Saldanha-da-Gama, 2014), it is necessary to manage and coordinate not just one process, but also multiple processes, taking place in the organisation, and multiple resources, shared by all the processes. That is why we have chosen to concentrate on the stability criterion. However, as the makespan is usually the most important or at least a criterion that can hardly be neglected, and it is in conflict with the stability criterion (Sabuncuoglu and Goren, 2009), we will take into account both criteria, and thus our approach will be a bicriteria approach to open shop scheduling.

The literature contains numerous concepts on the way to construct a stable initial schedule in several scheduling problems. All of them are listed in Sabuncuoglu and Goren (2009). Without entering into details, two statements can be formulated on the basis of Sabuncuoglu and Goren (2009) and other literature sources, known to the author:

1. There is no concept of a stable schedule that refers to the open shop problem. All the concepts of stable schedules refer to other types of scheduling problems (project schedules, flow shop, job shop). The uncertainty in the open shop problem has been taken into account, but this was done either through fuzzy modelling (see, e.g., Noori-Darvish, Mahdavi and Mahdavi-Amiri, 2012) or through stochastic modelling (see, e.g., Frostig, 1991);
2. The stability of a schedule is measured as:
   (a) the expected (mean) value, or
   (b) the variance, or
   (c) the worst case value

   of differences between selected features of the initial schedule (known in the decision-making moment) and the actual schedule (unknown in the decision-making moment, thus requiring assumptions). The selected features may be, for example the individual jobs starting times, the individual jobs completion times, the makespan, or the jobs order.

In this paper we will propose a concept of a stable schedule for the open shop problem, to our knowledge the first one in the literature so far. We will use a new measure of schedule stability: the certainty degree (assessed by experts) that the periods for which human resources are reserved for the process (known in the decision-making moment) and the actual time periods in which the resources will be used (unknown in the decision-making moment) will not differ. Our aim is to be able to plan the resources usage times in the organisation in such a way that possibly little changes will be necessary in their usage in the actual realisation. As mentioned above, we will combine this objective with that of a possibly small planned makespan.

Coming back to the state of the art, one of the tools used to achieve schedule stability in terms of deviations are time buffers. In machine scheduling, this term usually has another meaning (see, e.g., Zhang et al., 2013), but here we will use it in the sense adopted from project management (see, e.g., Herroelen and Leus, 2004). A buffer will thus be a period of idle time in the schedule, inserted at the beginning or at the end of a task/operation (or, in a more general case, before or after a group of tasks), often unknown to the task executors. It is not explicitly added to the estimated duration time of a task/operation, as psychologically this would push people to use it even if it were not really necessary (because of the so-called student syndrome, Goldratt, 1997). It is an extra reserve, known to the manager of the project or, in our case, process, which gives him more certainty that the whole project or process will not be disturbed with respect to the initial schedule and that it will not disturb other processes.

To our knowledge, the buffers as a tool ensuring schedule stability have been used only in project management (a recent application can be found in Kuchta, 2014), thus our approach is a novel proposal of applying it to machine scheduling, and, more exactly, to the open shop problem.

An open problem in each buffer application is the size of the buffers. There are many proposals in the area of project management; most of them are based on the features of individual activities or on project network features (see, e.g., Kobylański and Kuchta, 2007; Herroelen and Leus, 2004; Slusarczyk et al., 2013). In machine scheduling what will have to be taken into account while determining buffer size is:

- features of individual machines, jobs, operations;
- features of human resources responsible for jobs, machines, operations; and
- features of the ordered couples of jobs.

Some machines may require bigger buffers between jobs processed on them because they are less reliable than other machines. Some jobs or operations may be more innovative or complicated, or the responsible resources less experienced or competent, in which case the buffers should also be bigger. Some ordered couples of jobs may also require larger buffers between the two jobs: this is true, for example, if it is absolutely impossible to process both elements of the couple simultaneously using overtime. If the two jobs of the couple could be processed simultaneously, the simultaneous operation using extra hours could make up for the delays in the processing of the first job of the couple and the buffer between the two jobs may be smaller; if not, the buffer has to be bigger. The latter case may occur, for example, if a machine does the colouring of the jobs (elements being processed) and job i_o should be blackened and job i_1 whitened. The two jobs cannot possibly be simultaneously processed on this machine using overtime because the machine has to be cleaned very carefully after blackening; it is not possible to continue with blackening in extra time and simultaneously start whitening in normal time.

In our model the experts will be asked to formulate their requirements as to the individual buffer sizes. We want to insert at the end of each job enough time (a buffer), unknown to the job executor, such that we can be pretty sure that even if there are problems with job processing, the resources needed for the process and reserved for it, according to the initial schedule (the period of their reservation will be known only to the manager, not to the resources themselves), will be able to be used elsewhere outside these reservation periods. Thus, we want to minimize the differences between the periods, for which human resources needed in the process are initially planned, and the periods in which they will potentially be actually needed in the process in question. While aiming at the minimisation of these deviations, we will also aim for the minimisation of the planned makespan – for otherwise a very stable schedule with a very long makespan might be constructed, which would block an unnecessary amount of time, making the organisation lose time, money, and competitiveness, and the human resources lose motivation.

Given that we face two conflicting objectives, we apply a fuzzy approach: a decision maker will give for each buffer a lower limit of the buffer size, with which he would be fully satisfied, and an upper limit of the buffer size, with which he would be satisfied to the degree 0 – his degree of satisfaction between the two limits will be then expressed as a fuzzy number. Another decision maker will give an upper bound for the scheduled makespan, with which he would be completely satisfied, and a lower bound of the makespan values, where his satisfaction is equal to zero. The satisfaction of the decision maker with the scheduled makespan will be also modelled by means of a fuzzy number.

A mixed integer linear model will be formulated that will determine a schedule maximizing the minimum of all the satisfaction degrees of all the decision makers – the one interested in the stability of the schedule and the one interested in a small scheduled makespan. Let us indicate that a similar approach, but in the context of project management, has been applied in Kuchta (2014).

The structure of the paper is as follows: first, we present selected information about the type of fuzzy numbers that we use in our model. Then we describe an open shop model known from the literature that will form a basis for our model. Further, we formulate the problem using fuzzy numbers, and we finally transform this problem into a mixed integer linear model. An example is used to illustrate the proposal.

2. Basic information about fuzzy numbers

In this section we limit ourselves to the type of fuzzy numbers that we shall need in this paper.

Definition 1 *A left hand fuzzy number* $\tilde{A}^l = (\underline{a},\overline{a})^L$ *is determined by its membership function* $\mu_{\tilde{A}^l}:\mathrm{R} \longrightarrow [0,1]$ *where* R *is the set of real numbers, such that there exist two real numbers* $\underline{a}$ *and* $\overline{a}$, $\underline{a}\leqslant\overline{a}$, *such that* $\mu_{\tilde{A}^l}(x) = 1$ *for* $x\leqslant\underline{a}$, $\mu_{\tilde{A}^l}(x) = 0$ *for* $x\geqslant\overline{a}$ *and* $\mu_{\tilde{A}^l}$ *is continuous and decreasing in the interval* $(\underline{a},\overline{a})$*; and*
A right hand fuzzy number $\tilde{A}^r = (\underline{a},\overline{a})^R$ *is determined by its membership function* $\mu_{\tilde{A}^r}:\mathrm{R} \longrightarrow [0,1]$ *where* R *is the set of real numbers, such that there exist two real numbers* $\underline{a}$ *and* $\overline{a}$, $\underline{a}\leqslant\overline{a}$, *such that* $\mu_{\tilde{A}^r}(x) = 0$ *for* $x\leqslant\underline{a}$, $\mu_{\tilde{A}^r}(x) = 1$ *for* $x\geqslant\overline{a}$ *and* $\mu_{\tilde{A}^r}$ *is continuous and increasing in the interval* $(\underline{a},\overline{a})$.

Fuzzy numbers may have various interpretations, but in this paper we will use just one: for a given $x\in$ R, $\mu_{\tilde{A}^l}(x)$ or $\mu_{\tilde{A}^r}(x)$ represent the degree of satisfaction of a decision maker with a given value x. Various fuzzy numbers may represent the satisfaction of various decision makers, with various aspects, features or objects seen from various points of view. Left hand fuzzy numbers will be used in the case a decision maker would like a value to be rather small, and the right hand fuzzy numbers in the case when big values are preferred. The numbers $\underline{a}$ and $\overline{a}$ represent the limits of satisfaction – beyond the interval ($\underline{a}$,$\overline{a}$) only full or no satisfaction occurs, while inside the interval the intermediate satisfaction degrees are contained.

3. Formulation of the open shop problem with removal times

As a basis for our proposal we will use the slightly modified formulation of the open shop problem of Low and Yeh (2009): we have m machines and n jobs, p_{ij} (i= $1,\ldots,n$, $j = 1,\ldots,m$) stands for the processing time of the i-th job on the j-th machine. Each job has to be processed on each machine in any sequence. R_{ikj} ($i,k = 1,\ldots,n$, j= $1,\ldots,m$) stands for the necessary removal time of the i-th job before the immediate processing of the k-th job on the j-th machine. O_{ij} ($i = 1,\ldots,n$, $j = 1,\ldots,m$) stands for the processing of the i-th job on the j-th machine. $O_{n+1,j}$ – a dummy operation with processing time 0 that is assumed to be processed on machine j, $j = 1,\ldots,m$, when all the jobs have been processed and removed from this machine.

The removal time R_{ikj} ($i,k = 1,\ldots,n, j = 1,\ldots,m$) is treated in Low and Yeh (2009) as the time necessary to "clean up" the j-th machine between the completion of the i-th job and the start of the processing of the k^{th} job on machine j, if O_{kj} is scheduled immediately after O_{ij}. It is worth mentioning that, according to the model from Low and Yeh (2009), during this "cleaning up", the i-th job may already be processed on one of the other machines.

The following decision variables are used in the model from Low and Yeh (2009):

- y_{ijh} – equal to 1 if O_{ij} immediately precedes O_{ih}, $i = 1,\ldots,n$; $j,h = 1,\ldots,m$, equal to 0 otherwise;
- x_{lqj} – equal to 1 if O_{lj} precedes O_{qj}, $i,k = 1,\ldots,n$; $j = 1,\ldots,m$, equal to 0 otherwise;
- z_{lqj} – equal to 1 if O_{lj} immediately precedes O_{qj}, $i,k = 1,\ldots,n$; $j = 1,\ldots,m$, equal to 0 otherwise; and
- TS_{ij} – starting time of O_{ij} ($i = 1,\ldots,n, j = 1,\ldots,m$).

Apart from that, the following notation is used in the formulation of the constraints:

- A_j – set of operations that may be processed on machine j, $|A_j|$ denotes the cardinality of this set;
- E_j – set of operation pairs that may be processed on machine j, $j = 1,\ldots,m$;
- B_i – set of operation pairs for job i, $i = 1,\ldots,n$; and
- M – a large real number.

The objective function (minimised) will be the completion time of all the jobs on all the machines, taking into account the removal times. This completion time, in other words – the makespan, will be denoted as *MS*. Thus, we have:

$$MS = \left(TS_{ij} + p_{ij} + \sum_{a_{qj} \in A_j} R_{ikj} z_{ikj} \right). \tag{1}$$

The constraints are as follows:

$$\begin{gathered} TS_{ij} + p_{ij} - \mathrm{M}\,(1 - y_{ijh}) \leqslant TS_{ih} \quad TS_{ih} + p_{ih} - M y_{ijh} \leqslant TS_{ij} \\ \forall\ (O_{ij}, O_{ih}) \in B_i, \quad i = 1, \ldots, n \end{gathered} \tag{2}$$

$$\begin{gathered} TS_{ij} + p_{ij} + \sum_{a_{qj} \in A_j} R_{ikj} z_{ikj} - M\,(1 - x_{ikj}) \leqslant TS_{kj} \\ TS_{kj} + p_{kj} + \sum_{a_{qj} \in A_j} R_{ikj} z_{ikj} - M\,(1 - x_{ikj}) \leqslant TS_{ij} \\ \forall\ (O_{ij}, O_{kj}) \in E_j, \quad j = 1, \ldots, m \end{gathered} \tag{3}$$

$$\sum_{k=1}^{n+1} z_{kij} = 1, \quad \forall i, j \quad i = 1, \ldots, n; \quad j = 1, \ldots, m \tag{4}$$

$$x_{kij} + z_{kij} \geqslant 0,\ x_{kij} + z_{ikj} \leqslant 1 \quad \forall i, j, k \quad i, k = 1, \ldots, m; \quad j = 1, \ldots, m. \tag{5}$$

Constraints (2) express the relationships concerning the processing of one job on consecutive machines. Constraints (3) refer to individual machines and ensure the correctness of the order of the processing of individual jobs on the machines (here the necessary removal times are taken into account). Constraints (4) and (5) ensure correct values of decision variables x_{ikj} and z_{ikj} ($i, k = 1, \ldots, n; \quad j = 1, \ldots, m$), which both refer to the relationship between O_{ij} and O_{kj} ($i, k = 1, \ldots, n; \quad j = 1, \ldots, m$), but one takes into account just the fact or preceding or not, and the other one takes into account only immediate preceding.

Once the solution is determined, the values of the decision variables TS_{ij} $i = 1, \ldots, n$; $j = 1, \ldots, m$, will define the optimal schedule.

Let us illustrate the model, defined through formulae (1)–(5) with an example:

Example: Let us consider two machines and three jobs. The processing times are as in Table 1.:

The required removal times are assumed to be zero.

In this case we get: $TS_{11} = 0$, $TS_{12} = 7$, $TS_{21} = 8$, $TS_{22} = 1$, $TS_{31} = 3$, $TS_{32} = 8$, and the objective function value (the minimal makespan) is 9 time units.

Let us assume that for the processing of each job at any machine we need a human operator: person A for Job 1, person B for Job 2, person C for Job 3.

Table 1. Data for the Example

Job i	$\mathbf{p_{i1}}$	$\mathbf{p_{i2}}$
Job 1	2	1
Job 2	2	6
Job 3	5	2

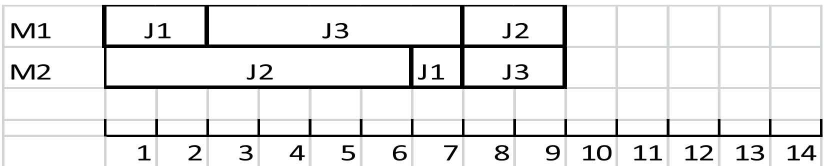

Figure 1. Optimal schedule for the Example

These persons are potentially also needed in other processes, which are realized in the organisation. If we accept for realisation the schedule from Fig. 1, we will reserve the resources for the periods given in Table 2.:

Outside these periods the resources will be considered to be free for other activities in the organisation.

Now, let us suppose that during the realisation of the schedule from Fig. 1 some problems occur with Job 1 on M1. As a result of these problems, the processing of this job on M1 takes 3 units instead of 2. Even if we assume that no other problems occur in the realisation of the schedule from Fig. 1, if there is no possibility to process Job 1 simultaneously with Job 3 on M1 in the 3 units of time (sometimes there might be such possibilities owing to the use of overtime hours, but, as mentioned in the introduction, this is not always the case), the schedule shown in Fig. 2 would be actually realised (the jobs marked grey are those, whose realisation time is different than in the planned schedule from Fig.1).

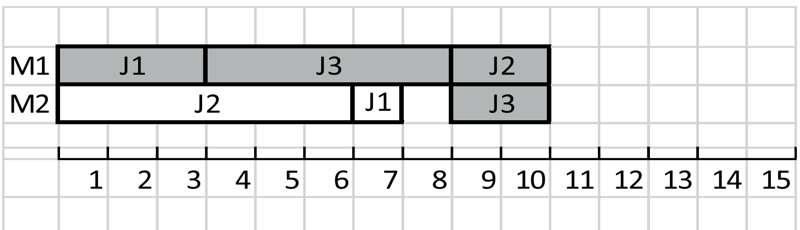

Figure 2. Actual schedule for the Example

The resources would, therefore, actually be needed for the process in question in slightly different periods than planned.

Table 2. Resource reservation periods for the Example

Resource	**Reservation period**
A	1-2, 7
B	1-6, 8-9
C	3-9

Table 3. Resource reservation and actual usage periods for the Example

Resource	**Reservation period**	**Actual usage period**	**Time units with potential problems for the whole organisation**
A	1-2, 7	1-3, 8	3, 8
B	1-6, 8-9	1-6, 9-10	10
C	3-9	4-10	10

In the time units, which are listed in the last column of Table 3, the resources A, B, C may have been planned for other processes in the organisation. If so, they will not be available for them as they still will be needed in the process in question due to the slight delay in finishing Job 1 on M1.

In the stable schedule, constructed according to the concept we propose here, we would like to fit the resources within the originally planned usage time even if disturbances occur. Hence, the aim is to have the last column of Table 3 possibly empty, so that problems causing changes in the jobs processing times of the process in question do not affect, or affect as little as possible, other processes in the organisation.

In the discussed example this would have been achieved if, having by some chance foreseen the possibility of problems with J1, we had inserted into the initial schedule a buffer after J1, just on M1 or on both machines. If we had thought the problems were job oriented and not machine oriented, we would have chosen to add a buffer after J1 on both machines, thus obtaining the schedule as shown in Fig.3.

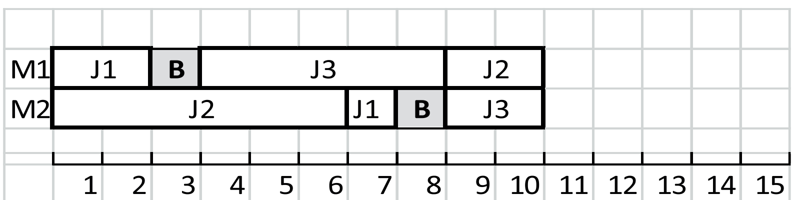

Figure 3. Planned schedule for the Example with buffers after job J1 operations

The buffers would not have been made known to resource A, in order to avoid the student syndrome problem. Thus, we would have had two human resources oriented schedules. One would have been known to the resources A, B, C, and it is given in Table 4.

Table 4. Resource schedule known to the resources, corresponding to Fig.3

Resource	**Assignment period**
A should do J1	1-2, 7
B should do J2	1-6, 9-10
C should do J3	4-10

Another schedule would be prepared for the management, as provided in Table 5.

Table 5. Resource schedule known to the management, corresponding to Fig.3

Resource	**Assignment period**
A	1-3, 7-8
B	1-6, 9-10
C	4-10

Owing to the usage of the buffer, the problem with the realisation of J1 described above (even accompanied by a similar problem with the same job on M2) would not have caused any disturbances in the organisation. In this sense, the schedule from Fig. 3 and Table 5 would have been stable – if only problems regarding J1 fitting into the buffers would have occurred.

Yet, of course, we prepared our stable schedule of Fig. 3 and Table 5 post factum. The aim is to have a model that would generate a stable schedule before process realisation. However, the stability of the schedule is not the only goal, because if it were so, we could have put huge buffers everywhere in the schedule and reserved the resource for the process in question for long periods, losing a lot of time and of opportunities, as the resources would be idle for a long time or would use the time inefficiently. So, the model should seek a compromise between two conflicting objectives: minimising the makespan of the schedule and maximising its stability. The respective model will be proposed in the following section.

4. Formulation of the stable schedule open shop problem

First of all, let us remark that we base our proposal on model (1)–(5), using the removal times as buffers. However, there will be two differences between model (1)–(5) and our approach:

- The removal times, called here buffers, will not be given as input data, but will be decision variables, whose value will have to be determined;

and

- During the removal time after the processing of a job on one machine, it will not be possible to process this job on any other machine (which is not the case in model (1)–(5)). This is how the buffers should act: they should reserve time for finishing the processing of a job on a machine in case of problems occurring during this processing and if problems do occur, the job will be blocked for processing on other machines. That is why constraints (2) have to be changed into:

$$\begin{aligned} & TS_{ij} + p_{ij} - \mathrm{M}\,(1 - y_{ijh}) + \sum_{a_{qj} \in A_j} R_{ikj} z_{ikj} \leqslant TS_{ih} \\ & TS_{ih} + p_{ih} - \mathrm{M} y_{ijh} + \sum_{a_{qj} \in A_j} R_{ikj} z_{ikj} \leqslant TS_{ij} \qquad (6) \\ & \forall\ (O_{ij}, O_{ih}) \in B_i, \quad i = 1, \ldots, n. \end{aligned}$$

In the proposed model we will use a fuzzy approach while modelling the preferences of various decision makers, who would have conflicting objectives. One decision maker would be interested in a possibly short makespan *MS*, in order not to block too much time in the organisation for the process in question. He would express his preferences in the form of a left-hand fuzzy number $\tilde{MS}^l = (\underline{ms}, \overline{ms})^L$, with the membership function $\mu_{\tilde{MS}^l}$.

Another decision maker would be interested in a high stability of the schedule, so that problems in its realisation would possibly not affect the rest of the organisation.

The latter type of decision maker would formulate his views or preferences on the basis of, for example:

- The proneness of individual machines to break down (if a machine is more suspect to fail, the buffers on this machine should be higher);
- The experience and reliability of human resources assigned to the process (less experienced and less reliable resources require longer buffers);
- The characteristics of individual jobs (jobs which are more complicated or more unusual for the organisation require longer buffers); and
- The characteristics of job couples, related to machines:
  - some jobs may, in case of a delay on a machine, be processed on this machine simultaneously with another job, thanks to, for example overtime – for such job couples shorter buffers between the jobs will be sufficient;
  - some job couples cannot be processed simultaneously on the same machine (an example was given in the introduction) – they will need longer buffers between jobs.

We will model the preferences of such a decision maker using right hand fuzzy numbers $\tilde{R_{ikj}}^r = \left(\underline{r}_{ikj}, \overline{r}_{ikj}\right)^R$ with membership functions $\mu_{\tilde{R_{ikj}}^r}$. In some cases we may have identical $\tilde{R_{ikj}}^r$ for a fixed j, if the necessary buffers depend only on the machine and its proneness to failure. In other special cases we may have identical $\tilde{R_{ikj}}^r$ for a fixed i, if the buffer depends only on the job characteristics. In yet other cases we may have special values for selected couples (i,k) of jobs coming one after another, sometimes for selected triples (i,j,k).

The aim will be to maximise the total satisfaction of all the decision makers, measured as:

$$min\left\{\mu_{\tilde{MS}^l}\left(MS\right), min_{i,k,j}\mu_{\tilde{R_{ikj}}^r}\left(R_{ikj}\right)\right\} \rightarrow max \tag{7}$$

with constraints (3)–(6).

The fuzzy number $\mu_{\tilde{R_{ikj}}^r}\left(R_{ikj}\right)$ $(\forall i,k = 1,\dots,n; \quad j = 1,\dots,m)$ measures the degree of possibility that the usage of resources assigned to operation O_{ij} (like in the Example, where resource A was assigned to operations O_{11} i O_{12}), in case when on the j-th machine immediately after the i-th job the k-th job is processed, will not exceed $p_{ij}+R_{ikj}$. Thus, $\mu_{\tilde{R_{ikj}}^r}\left(R_{ikj}\right)$ is a measure of stability of the schedule of resources assigned to the open shop problem. The other component of the objective function (7) measures the satisfaction of the decision maker with the planned makespan.

Problem (3)–(7) can be expressed as a linear programming problem if the membership functions measuring the satisfaction of the decision makers are piecewise linear, what we assume here. The linear programming problem is then as follows:

$$\sigma \rightarrow max \tag{8}$$

$$TS_{ij} + p_{ij} + \sum_{a_{qj} \in A_j} R_{ikj} z_{ikj} \leqslant \lambda \ (\forall i,j,k : i,k = 1,\dots,n; \ j = 1,\dots,m) \tag{9}$$

$$\underline{ms} \leqslant \lambda \leqslant \overline{ms} \tag{10}$$

$$\mu_{\tilde{MS}^l}\left(\lambda\right) \geqslant \sigma \tag{11}$$

$$\underline{r_{ikj}} \leqslant R_{ikj} \leqslant \overline{r}_{ikj} \ (\forall i,j,k : \quad i,k = 1,\dots,n; \quad j = 1,\dots,m) \tag{12}$$

$$\mu_{\tilde{R_{ikj}}^r}\left(R_{ikj}\right) \geqslant \sigma \ (\forall i,j,k : \quad i,k = 1,\dots,n; \quad j = 1,\dots,m) \tag{13}$$

All the constraints of the above problem are linear, under the assumption that the membership functions are piecewise linear. It is enough to consider the intervals where the membership functions increase from 0 to 1 or decrease from 1 to 0 (constraints (10) and (12)), because the maximum value of the membership function σ will certainly be found there. There may be a better solution than the one found in this way – better in terms of the buffer size or the makespan – but not in terms of the satisfaction of the decision makers, and we are interested only in the decision maker's satisfaction measured by the fuzzy numbers given by them.

As a result of solving model (8)–(13), we will get a schedule with buffers, ensuring the highest possible total satisfaction of both the decision maker concerned with the stability of the schedule and the decision maker concerned with a possibly short makespan.

Let us consider the following variation of the Example: the data from Table 1 are adopted, plus the requirements of the decision makers:

- regarding the makespan: $\tilde{MS}^{l} = (12,\ 14)^{L}$
- regarding schedule stability, thus the buffers: $\tilde{R_{ikj}}^{r} = (0,2)$ for all i,k =1,2,3, j =1,2, except for i = 1 and k = 3, $\tilde{R_{13j}}^{r}$ =$(2,5)$ for $j = 1,2$.

The sequence J1, J3 requires, therefore, an especially big buffer between the two jobs on both machines.

The optimal solution of model (8)–(13) for this data includes the optimal objective function (8) value of 0.5, equal to that of the objective function (7), being the maximal total satisfaction of all the decision makers, and the schedule as given in Fig. 4.

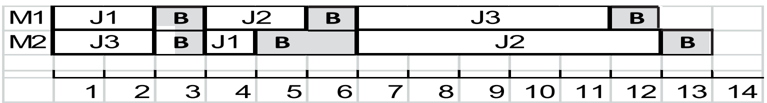

Figure 4. Optimal stable schedule for the modified Example

In the case of the schedule from Fig. 4, the decision maker interested in a possibly small makespan is satisfied to the degree 0.5 and so is the decision maker concerned about schedule stability. According to this schedule, we would make a reservation of the resources for the periods, specified in Table 6.

Table 6. Stable resource schedule known to the management, corresponding to Fig.4

Resource	**Reservation period**
A	1-5
B	4-12
C	1-2, 6-11

If we took the initial optimal schedule for the Example (Fig. 1), which did not take into account the stability problem, and if we simply inserted after each operation the buffer corresponding to the satisfaction degree 0.5, we would get the schedule as shown in Fig. 5.

The satisfaction degree of the planned makespan of the schedule from Fig. 5 is 0.33, which would also be the value of the total satisfaction (7). Any attempt

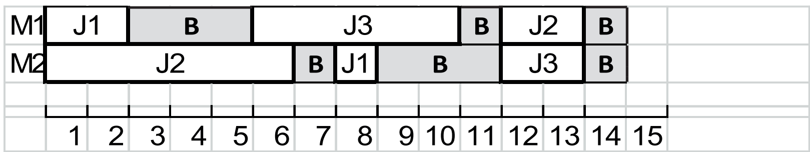

Figure 5. Optimal schedule for the Example from Fig. 1, made stable with the satisfaction degree 0.5 without changing the operation order

to improve this satisfaction degree without changing the order of operations would decrease satisfaction with schedule stability. The application of model (3)–(7) has allowed for avoiding the couple J1, J3 as immediate neighbours in processing (with J1 being processed at first) and, owing to this, to achieve the total satisfaction degree equal 0.5.

5. Conclusions

In this paper we propose the first – to our knowledge – open shop schedule stability concept. Stability is here understood in terms of ensuring that the initial resource usage schedule will differ as little as possible from the actual one. The tools used in our model to guarantee schedule stability are constituted by the time buffers. The buffers inserted into the schedule permit to protect the whole organisation against problems due to delays in various processes executed in the organisation and the resulting resource demand conflicts. Fuzzy modelling is used to express the compromise between the need for a possibly stable and for a possibly short schedule, but finally an equivalent standard mixed integer linear programming problem has to be solved in order to find a solution.

The problem of a stable open shop schedule should be further exploited, as this aspect seems to have been neglected in the literature so far. New concepts should be proposed and verified in practice, and this will constitute the future research directions.

Acknowledgement

This work is supported by the Polish National Science Centre grant “The buffer size and location in proactive and reactive risk management of project duration overrun” 2011/01/B/HS4/02821.

References

ALOULOU, M.A. and ARTIGUES, C. (2010) Flexible solutions in disjunctive scheduling: General formulation and study of the flow-shop case. *Computers & Operations Research,* **37**(5), 890-898.

BAKER, K.R. and TRIETSCH, D. (2009) *Principles of Sequencing and Scheduling.* John Wiley & Sons, Inc.

CORREIA, I. and SALDANHA-DA-GAMA, F. (2014) The impact of fixed and variable costs in a multi-skill project scheduling problem: An empirical study. *Computers & Industrial Engineering,* **72**(0), 230-238.

DONG, J., ZHANG, A., CHEN, Y. and YANG, Q. (2013) Approximation algorithms for two-machine open shop scheduling with batch and delivery coordination. *Theoretical Computer Science,* **491**(0), 94-102.

FROSTIG, E. (1991) On the optimality of static policy in stochastic open shop. *Operations Research Letters,* **10**(9), 509-512.

GOLDRATT, E.M. (1997) *Critical Chain.* North River Press.

HERROELEN, W. and LEUS, R. (2004) The construction of stable project baseline schedules. *European Journal of Operational Research,* **156**(3), 550-565.

JENSEN, M.T. (2001) Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures. *Applied Soft Computing,* **1**(1), 35-52.

KASPERSKI, A., KURPISZ, A. and ZIELIŃSKI, P. (2012) Approximating a two-machine flow shop scheduling under discrete scenario uncertainty. *European Journal of Operational Research,* **217**(1), 36-43.

KOBYLAŃSKI, P. and KUCHTA, D. (2007) A note on the paper by M. A. Al-Fawzan and M. Haouari about a bi-objective problem for robust resource-constrained project scheduling. *International Journal of Production Economics,* **107**(2), 496-501.

KUCHTA, D. (2014) A new concept of project robust schedule – Use of buffers. *Procedia Computer Science,* **31**(0), 957-965.

LOW, C. and YEH, Y. (2009) Genetic algorithm-based heuristics for an open shop scheduling problem with setup, processing, and removal times separated. *Robotics and Computer-Integrated Manufacturing,* **25**(2), 314-322.

NOORI-DARVISH, S., MAHDAVI, I. and MAHDAVI-AMIRI, N. (2012) A bi-objective possibilistic programming model for open shop scheduling problems with sequence-dependent setup times, fuzzy processing times, and fuzzy due dates. *Applied Soft Computing,* **12**(4), 1399-1416.

PANWALKAR, S.S. and KOULAMAS, C. (2014) The two-machine no-wait general and proportionate open shop makespan problem. *European Journal of Operational Research,* **238**(2), 471-475.

SABUNCUOGLU, I. and GOREN, S. (2009) Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. *International Journal of Computer Integrated Manufacturing,* **22**(2), 138-157.

SLUSARCZYK, A., KUCHTA, D., VERLHUST, P., HUYGHE, W., LAURYSSEN, K. and DEBAL, T. (2013) A comparison of buffer sizing techniques in the critical chain method. Case study. *Journal of Automation, Mobile Robotics & Intelligent System,* **7**(3), 43-56.

XIONG, J., XING, L. and CHEN, Y. (2013) Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. *International Journal of Production Economics,* **141**(1), 112-126.

ZHANG, P.-w., PAN, Q.-k., LI, J.-q. and DUAN, J.-h. (2013) Hybrid artificial bee colony algorithms for flowshop scheduling problem with limited buffers. *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems, CIMS,* **19**(10), 2510-2520.