Volume 23 (1), pp. 17–28
http://www.orssa.org.za

*ORiON*
ISSN 0529-191-X


# Selection and scheduling of jobs with time-dependent duration

DM Seegmuller[†] SE Visagie[*] HC de Kock[†] WJ Pienaar[†]

*Received: 6 October 2006; Revised: 11 January 2007; Accepted: 13 March 2007*

**Dedication to Emeritus Professor Gerhard Geldenhuys**

*Gerhard Geldenhuys is probably the person who left the largest footprint in teaching Operations Research (OR) in South Africa. The fact that three of the authors received the vast majority of their OR education from him, supports this view. All four the authors, spanning more than one generation, were shaped or influenced by his enthusiasm about OR and his sincere interest in his students. It is with great respect and appreciation that we dedicate this paper to him. Dankie, Gerhard, vir jou groot bydrae.*

**Abstract**

In this paper two mathematical programming models, both with multiple objective functions, are proposed to solve four related categories of job scheduling problems. All four of these categories have the property that the duration of the jobs is dependent on the time of implementation and in some cases the preceding job. Furthermore, some jobs (restricted to subsets of the total pool of jobs) can, to different extents, run in parallel. In addition, not all the jobs need necessarily be implemented during the given time period.

**Key words:** Scheduling, sequencing, 0/1 integer programming, time dependence of job duration.

## 1 Introduction

Over the last few decades a substantial amount of research has been done in the field of job scheduling where the duration of the jobs is time dependent [1, 6]. A vast number of algorithms and models have been developed to represent and solve a wide variety of real life problems arising form production and manufacturing. Notwithstanding a loss of generality, most of the algorithms developed to solve the problem above are more efficient than the equivalent algorithms based on conventional integer programming models.

[*]Corresponding author: Department of Logistics, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa, email: svisagie@sun.ac.za

[†]Department of Logistics, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa

The problem considered in this paper is concerned with determining an optimal sequence of tests to be performed at a specific test station. Individuals wanting to use the test station must apply to occupy the test station and sometimes may even choose the time at which to carry out the test. This results in a pool of tests and the management of the test station must select a subset of these tests and determine a sequence in which the tests should be performed in order to maximise income. In this case, not only is the test duration dependent on the start times, but may also depend on the preceding test as well. Furthermore, in certain circumstances parts of tests can run in parallel if they require the same equipment setup.

This problem may be described as a multi-objective scheduling problem with time dependent duration and in which jobs can run in parallel to a certain degree. The first objective is to select the right jobs to implement from a pool of waiting jobs and the second is to minimise the total duration of the jobs selected. No models could be found in the literature to solve exactly the problem formulation as stated above.

In the next section a brief background to the problem and literature is given. In §3 the relevant mathematical models are provided to solve the different occurrences of the scheduling problem. These models are followed in §4 by a few notes on the complexity of the algorithms. Examples to demonstrate the implementation of the model are given in §5. In §6 conclusions are drawn and ideas for further research are suggested.

## 2 Background

Pinedo [14] classifies scheduling problems according to a triplet notation $\alpha|\beta|\gamma$, where $\alpha$ describes the machine environment, $\beta$ the processing characteristics and constraints, and $\gamma$ the objective to be optimised. According to this notation, the machine environment for our problem is a single machine setup (denoted by 1). The processing characteristics can be pre-emptive[1] ($prmp$) with sequence-dependent setup times ($s_{j,k}$) and precedence ($prec$) restrictions. Finally, there are two objectives, namely, the maximisation of the total weighted jobs ($\sum w_j$) and the minimisation of makespan ($C_{\max}$). Thus in triplet notation our problem is related to a $1|prmp, s_{j,k}, prec|\sum w_j, C_{\max}$ scheduling problem.

Generally, objective functions considered in the scheduling literature [1, 6, 14] are based on time measures, such as the maximisation or minimisation of one or a combination of makespan $C_{\max}$, maximum lateness $L_{\max}$, maximum tardiness $T_{\max}$ and variance of makespan $var(C_{\max})$. On the contrary, our problem is not purely pre-emptive and does not include any time measure. Consequently, it is not possible to solve our problem by means of existing algorithms. Alternatively, one could formulate the problem as a 0/1 integer programming problem, but the resulting model is complex. Cheng *et al.* [6] state that almost all scheduling problems where the duration of jobs is time dependent are NP-complete.

Another difference between the problem considered here and the general problems in the literature, where durations of jobs depends on their start times, is that the models in the

[1] In our problem description jobs are pre-emptive with other jobs having a similar equipment setup configuration, which is equivalent to jobs running in parallel that are in the same setup configuration class.

literature assume some functional relationship between the duration of a job and its start time [6]. The suggested functional relationships are:

1. The duration is either increasing or decreasing as the start time increases. The functional relationship is then given by $d_i = a_i \pm b_i s_i$, where $d_i$ is the duration, $s_i$ the start time of job $i$, and $a_i$ and $b_i$ are constants. See, for example, [2, 4, 7, 8] for articles using this approach.
2. The duration is either constant and then increasing ($d_i = \max\{a_i, a_i + b_i(s_i - p_i)\}$) or decreasing and then constant ($d_i = a_i - b_i \min(s_i, p_i)$) as the start time increases. The same notation as above is used, with the addition that $p_i$ is a constant too. See, for example, [5, 11] for this approach.
3. The duration is a step function ($d_i = a_i$ or $a_i + b_i$). See, for example, [3, 9] for this approach.

No such functional relationships exist in the situation considered here. In our application the durations can, for example, be constant, increase, decrease and increase again.

## 3 Mathematical models

The objective in this paper is to formulate the selection and scheduling of jobs with time dependent duration as an integer program (IP) with multiple objective functions. Cases are considered where the job duration and cost are dependent on the time and sequence of job implementation. The pool of jobs is restricted to a predetermined set and the set of time periods is restricted to integer values. The job duration functions are restricted to cases where the job duration can be predetermined for a given start time. The discreteness of the job durations and start times do not impose restrictions on the applicability of the models. If necessary, the models can accommodate any length of interval by simply choosing smaller intervals (for example, hours instead of days or weeks).

Four scheduling models which model different degrees of job duration and overlap, and therefore different degrees of complexity, are considered. Problems are classified in the following four categories:

**Category 1:** The duration of a job depends on its start time and jobs cannot run in parallel.

**Category 2:** The duration of a job depends on its start time and jobs can run in parallel.

**Category 3:** The duration of a job depends on its start time and on the preceding job, and jobs cannot run in parallel.

**Category 4:** The duration of a job depends on its start time and on the preceding job, and jobs can run in parallel.

An optimal job sequence should be determined according to the two objectives (in order of priority) given below:

1. Maximise the profit generated by jobs that are implemented; and

2. Minimise the total duration of the jobs that are implemented.

Only two mathematical models are necessary to model the four categories. The value of one parameter determines whether or not jobs can run in parallel as well the extent to which jobs can run in parallel. The two model formulations follow in the sections below.

### 3.1 Job duration depends on the start time only

Categories 1 and 2 are considered first. In both these categories precedence is not important and will be modelled by means of an IP. A description of the variables required in modelling the case where the job duration dependents on the start times is given below:

$d_{i,k}$ duration of job $i$ when implemented during period $k$,
$c_{i,k}$ profit margin of job $i$ when implemented during period $k$,
$h_i$ parameter indicating the fraction of the facility that job $i$ occupies for the full duration of implementation,
$r_i$ relative weight representing the relative importance of job $i$,
$x_{i,k}$ binary variable equal to unity if the implementation of job $i$ starts during period $k$, and
$b_{i,k}$ binary variable equal to unity if job $i$ is implemented during period $k$.

The following assumptions are made in the model:

1. Jobs run continuously and must be completed within the given time intervals $1, \ldots, w$.

2. The durations of jobs have integer values.

3. A discrete range of values is assigned to $h_i$, for example, $h_i = [1, 0.75, 0.5, 0.25]$.

4. Jobs do not require a specific equipment/facility setup, which implies that any subset of jobs can run in parallel, if the resources permit.

5. The durations of jobs that can run in parallel are defined for a specific value of $h_i$. Usually, jobs running one after the other will imply shorter individual job durations and therefore separate jobs are defined with relative durations and cost functions for the range of valid $h_i$ values. The set of derived jobs is denoted by $\mathcal{H}_i$. Take, for example, job $J_1$ which has a duration of 4 units if it is run on its own. If it can run in parallel with $h_1$ taking on the values $1, 0.75, 0.5$ and $0.25$, then the set of four derived jobs $\mathcal{H}_1 = \{J_{1a}, J_{1b}, J_{1c}, J_{1d}\}$ with durations of 4, 6, 7 and 8 respectively are created. The cost function of the additional jobs should reflect a saving in cost in the overheads, since running jobs in parallel generally decreases the total duration and costs to complete all the jobs. Only one of these derived jobs will be implemented.

6. Maximisation of the weighted number of jobs has priority over the minimisation of total job durations and will be optimised first.

A mathematical formulation for the case where jobs can run in parallel is presented. The case where jobs cannot run in parallel may be treated as a special case by simply setting the parameter $h_i = 1$ for all jobs. With reference to the variables defined, the IP formulation is given by

$$\max \sum_{i=1}^{n} \sum_{k=1}^{w} r_i c_{i,k} x_{i,k} \tag{1}$$

and

$$\min \sum_{i=1}^{n} \sum_{k=1}^{w} d_{i,k} \tag{2}$$

subject to

$$\sum_{k=1}^{w} x_{i,k} \leq 1, \qquad i = 1, \ldots, n \tag{3}$$

$$\sum_{i=1}^{n} h_i b_{i,k} \leq 1, \qquad k = 1, \ldots, w \tag{4}$$

$$\sum_{\substack{\ell=1 \\ \ell \geq k}}^{k+d_{i,k}-1} b_{i,\ell} \geq d_{i,k} x_{i,k}, \qquad i = 1, \ldots, n, \quad k = 1, \ldots, w \tag{5}$$

$$(w-1)x_{i,k} + d_{i,k} \leq 2w - k, \qquad i = 1, \ldots, n, \quad k = 1, \ldots, w \tag{6}$$

$$x_{s,k} = 0, \qquad \text{for all } (s,k) \in \mathcal{S}_k \tag{7}$$

$$\sum_{k=1}^{w} \sum_{m \in \mathcal{H}_i} x_{m,k} \leq 1, \qquad \text{for all } \mathcal{H}_i \tag{8}$$

$$x_{i,k}, b_{i,k} \in \{0,1\}, \qquad i = 1, \ldots, n, \quad k = 1, \ldots, w \tag{9}$$

$$d_{i,k} \in \mathbb{Z}, \qquad i = 1, \ldots, n, \quad k = 1, \ldots, w. \tag{10}$$

Constraint (3) ensures that a job starts once only. If the value of $h_i$ is less than 1 it means that jobs can run in parallel. Constraint (4) dictates how many jobs can run in parallel during a certain time period, while not exceeding the capacity of equipment or the facility. Constraint (5) ensures that the binary variables $b_{i,k}$ take on a value of one for all the time periodes $k$ during which job $i$ is running. Constraint (6) states that if $k + d_{i,k} - 1 > w$, then $x_{i,k} = 0$. In other words, it ensures that a job can only start if it can be finished before the end of the time periods over which the model is solved. Constraint (6) is thus a simplification of

$$k + d_{i,j} - 1 - w \leq M(1 - x_{i,j}),$$

where $M$ is a large positive number. If it is assumed that $\max_{i,k}\{d_{i,k}\} < w$, then $M = w-1$ is sufficiently large. Constraint (7) prohibits job $s$ from starting during time period $k$. The set $\mathcal{S}_k$ contains all the jobs that are not allowed to start at the beginning of time period $k$. The final constraint in equation (8) ensures that at most one element from the set of derived jobs $\mathcal{H}_i$ can be chosen.

### 3.2 Job duration depends on the start time and the preceding job

Many applications arise where job execution requires a specific setup of the facilities and resources. In this section we consider the situation where the setup times for job execution depend on the state of the facility after the preceding job has been completed. The sequence thus plays an integral part in the duration and cost of implementing jobs. Since jobs require specific facility setups, not all jobs can run in parallel. Jobs should therefore be categorised into classes of similar facility setup requirements with similar setup durations and cost portions of the total job cost. This case is again formulated as an IP and the case where jobs cannot run in parallel is a special case obtained by simply setting the parameter $h_i$ equal to unity for all jobs. In addition the following assumptions are added to those in §3.1:

1. Jobs may be categorised into classes of similar facility requirements, with similar setup cost portions of the total job cost and similar setup duration portions of the total duration required for job implementation.

2. The facilities are in an initial setup state before the commencement of the first period.

3. Only jobs within the same setup category can run in parallel.

The description of the variables required in modelling the case where the job duration depends on the start times is given below:

| | |
|---|---|
| $x_{i,j,k}$ | binary variable equal to unity if the implementation of job $i$ starts during period $k$ and follows on job $j$, |
| $d_{i,j,k}$ | duration of job $i$ if it follows on job $j$ and is implemented during period $k$, |
| $c_{i,j,k}$ | profit if job $i$ is implemented during period $k$ and follows on job $j$, |
| $h_i$ | parameter indicating the fraction of the facility that job $i$ occupies for the full duration of implementation, |
| $r_i$ | relative weight representing the relative importance of job $i$, and |
| $b_{i,k}$ | binary variable equal to unity if job $i$ is implemented during period $k$. |

In terms of the notation above, the model formulation for categories 3 and 4 is

$$\max \sum_{i=1}^{n} \sum_{k=1}^{w} r_i c_{i,j,k} x_{i,j,k} \tag{11}$$

and

$$\min \sum_{i=1}^{n} \sum_{k=1}^{w} d_{i,j,k} \tag{12}$$

subject to

$$\sum_{\substack{j=1\\j\neq i}}^{n}\sum_{k=1}^{w} x_{i,j,k} \leq 1, \qquad i = 1,\ldots,n \tag{13}$$

$$\sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq i}}^{n} h_i b_{i,k} \leq 1, \qquad k = 1,\ldots,w \tag{14}$$

$$\sum_{\substack{\ell=1\\\ell\geq k}}^{k+d_{i,k}-1} b_{i,\ell} \geq d_{i,j,k}x_{i,j,k}, \quad i = 1,\ldots,n, \quad j = 1,\ldots,n, \quad k = 1,\ldots,w \tag{15}$$

$$(w-1)x_{i,j,k} + d_{i,j,k} \leq 2w - k, \quad i = 1,\ldots,n, \quad j = 1,\ldots,n, \quad k = 1,\ldots,w \tag{16}$$

$$\sum_{\substack{i=1\\i\neq j}}^{n} x_{i,j,k} - \sum_{\substack{i=1\\i\neq j}}^{n}\sum_{\ell=1}^{k-1} x_{i,j,\ell} \leq 0, \quad j = 1,\ldots,n, \quad k = 2,\ldots,w \tag{17}$$

$$x_{s,j,k} = 0, \qquad \text{for all } (s,j,k) \in \mathcal{S}_k \tag{18}$$

$$\sum_{k=1}^{w}\sum_{m\in\mathcal{H}_i}\sum_{j=1}^{n} x_{m,j,k} \leq 1, \qquad \text{for all } \mathcal{H}_i \tag{19}$$

$$x_{i,j,k}, b_{i,k} \in \{0,1\}, \quad i = 1,\ldots,n, \quad j = 1,\ldots,n, \quad k = 1,\ldots,w \tag{20}$$

$$d_{i,j,k} \in \mathbb{Z}, \quad i = 1,\ldots,n, \quad j = 1,\ldots,n, \quad k = 1,\ldots,w. \tag{21}$$

The goal given by (11) maximises the weighted number of chosen jobs, while the goal given in (12) minimises the total duration of these jobs. The constraints are similar to the ones used in §3.1, with generalisation for the precedence complications. The only constraint not present in the formulation in §3.1 is constraint (17). This constraint is also an if-then constraint and states that if job $i$ is preceded by job $j$ and starts during time period $k$, then job $j$ should have started before time period $k$.

## 4 Complexity of models

The size of the IP is a function of the number of jobs in the pool, the number of time periods, the problem classification and problem-specific limitations. Problem-specific limitations may involve the following:

1. The refinement of a unit period interval into fractions of a period, which defines the range of valid $h_i$ values. For example, when refining the unit interval to quarter units and specifying a job with an $h_i$ value of 1, $\frac{3}{4}$, $\frac{1}{2}$ and $\frac{1}{4}$ implies that 3 new jobs must be defined with valid cost and duration parameters. This increases the job pool to select from.

2. For category 3 and category 4 classifications, jobs are classified according to facility configuration requirements. Jobs within different classes cannot run in parallel.

Table 1 gives a summary of the order of magnitude of the size of a typical scheduling problem solved by the IP models discussed above. The data in this table are based on a model where there are a total of $n$ jobs including derived jobs and a total of $w$ time periods. From Table 1 it may be seen that the complexity increases as the number of jobs and periods increases. The most serious problem, however, is not the number of constraints but the number of 0/1 variables which blow up the solution times unacceptably.

There are problem-dependent complexities, such as the ones reflected in constraints (7), (8), (18) and (19), that should also be taken into account and that are not reflected in Table 1. All four of the categories can become more complex if there are time preferences, *i.e.* certain jobs are strongly preferred to be completed during certain time periods. Categories 2 and 4 become more complex with an increase in the number of jobs that can run in parallel. Categories 3 and 4 become more complex with an increase in the number of job classes.

| Fomulation | Category 1 & 2 | Category 3 & 4 |
|---|---|---|
| Parameters | $n(2w+2)$ | $2n(wn+1)$ |
| Variables | $2nw$ | $nw(n+1)$ |
| Constraints | $n(2w+1)+w$ | $nw(2n+1)+w+n$ |

**Table 1:** *Summary of the order of magnitude of the number of variables and constraints for the job scheduling models in §3.1 and §3.2.*

The different formulations were coded for, and implemented in LINGO [12, 13]. Formulations for real problems were solved using LINGO, but solution times are unrealistically long. Typical solution times for practical problems are in the order of one to three days.

## 5 Examples

The following examples demonstrate the implementation of the different models. The examples were solved with LINGO. The four different categories discussed in §3 are considered for the example.

Consider a pool of jobs $\{J_1, J_2, J_3, J_4, J_5, J_6, J_7, J_8\}$ that may be implemented in the time periods $0, \ldots, 12$. The duration ($D$) in months and profit ($P$) in thousands of rands for categories 1 and 3 (*i.e.* jobs cannot run in parallel) are

$$D = \begin{array}{c} \\ J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_6 \\ J_7 \\ J_8 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\ \left[\begin{array}{cccccccccccc} 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 & 5 & 5 & 5 & 5 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 1 & 1 \\ 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 5 & 5 & 5 & 4 & 4 & 4 & 4 \\ 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] \end{array}$$

and

$$P = \begin{array}{c} \\ J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_6 \\ J_7 \\ J_8 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\ \left[\begin{array}{cccccccccccc} 35 & 35 & 35 & 35 & 35 & 40 & 40 & 40 & 35 & 35 & 35 & 35 \\ 28 & 28 & 28 & 25 & 25 & 25 & 25 & 25 & 25 & 25 & 25 & 25 \\ 35 & 35 & 35 & 35 & 38 & 38 & 38 & 38 & 38 & 35 & 35 & 35 \\ 9 & 9 & 9 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 9 & 9 \\ 22 & 22 & 22 & 23 & 23 & 23 & 23 & 23 & 22 & 22 & 22 & 22 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 30 & 30 & 30 & 30 & 32 & 32 & 32 & 32 & 32 & 32 & 30 & 30 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right] \end{array},$$

where $d_{ik}$ is the duration if job $i$ if it starts during period $k$, and $p_{ik}$ is the profit of job $i$ if it starts during time period $k$.

The set of jobs $\{J_1, J_6, J_7\}$ can run in parallel with corresponding $h_i$ values of 0.5 for all of them. The artificially created jobs that can run in parallel will be called $\{J_{1a}, J_{6a}, J_{7a}, J_{1b}, J_{6b}, J_{7b}\}$. The duration ($D$) in months and profit ($P$) in thousands of rand for categories 2 and 4 (*i.e.* jobs can run in parallel) are

$$D = \begin{array}{c} \\ J_{1a} \\ J_{1b} \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_{6a} \\ J_{6b} \\ J_{7a} \\ J_{7b} \\ J_8 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\ \left[\begin{array}{cccccccccccc} 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 & 5 & 7 & 7 & 7 & 5 & 5 & 5 & 5 \\ 3 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 & 5 & 5 & 5 & 5 \\ 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 1 & 1 \\ 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 5 & 5 & 5 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 & 7 & 7 & 7 & 5 & 5 & 5 & 5 \\ 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 2 & 2 \\ 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] \end{array}$$

and

$$P = \begin{array}{c} \\ J_{1a} \\ J_{1b} \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_{6a} \\ J_{6b} \\ J_{7a} \\ J_{7b} \\ J_8 \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{array} \\ \left[\begin{array}{cccccccccccc} 35 & 35 & 35 & 35 & 35 & 40 & 40 & 40 & 35 & 35 & 35 & 35 \\ 40 & 40 & 40 & 40 & 40 & 45 & 45 & 45 & 40 & 40 & 40 & 40 \\ 28 & 28 & 28 & 25 & 25 & 25 & 25 & 25 & 25 & 25 & 25 & 25 \\ 35 & 35 & 35 & 35 & 38 & 38 & 38 & 38 & 38 & 35 & 35 & 35 \\ 9 & 9 & 9 & 11 & 11 & 11 & 11 & 11 & 11 & 11 & 9 & 9 \\ 22 & 22 & 22 & 23 & 23 & 23 & 23 & 23 & 22 & 22 & 22 & 22 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 30 & 30 & 30 & 30 & 32 & 32 & 32 & 32 & 32 & 32 & 30 & 30 \\ 30 & 30 & 30 & 30 & 32 & 32 & 32 & 32 & 32 & 32 & 30 & 30 \\ 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \end{array}\right] \end{array}.$$

Additional data are required in categories 3 and 4. Let there be three different configuration setup subsets, namely class 1, class 2 and class 3. These classes consist of the following jobs: class 1 = $\{J_1, J_3, J_4, J_6\}$, class 2 = $\{J_2, J_7, J_8\}$ and class 3 = $\{J_5\}$. The jobs in class 2 require 1 day additional setup time if preceded by jobs in class 1, with an additional setup cost of R5 000. The setup times ($S$) due to necessary adjustments to equipment and the cost associated with the adjustments ($C$) are

$$S = \begin{array}{c} \\ J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_6 \\ J_7 \\ J_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 & J_8 \end{array} \\ \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}\right] \end{array}$$

and

$$C = \begin{array}{c} \\ J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_6 \\ J_7 \\ J_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} J_1 & J_2 & J_3 & J_4 & J_5 & J_6 & J_7 & J_8 \end{array} \\ \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 5 & 5 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 5 & 5 & 0 & 5 & 0 & 0 \\ 5 & 0 & 5 & 5 & 0 & 5 & 0 & 0 \end{array}\right] \end{array},$$

where $s_{ij}$ is the additional setup time for job $i$ if job $j$ precedes it and $c_{ij}$ is additional cost added to job $i$ if job $j$ precedes it. If jobs can run in parallel, more rows and columns are added to $S$ and $C$, to supply the values for all the derived jobs in the set(s) $\mathcal{H}_i$. The values in these additional rows and columns stay the same as values in the original row or column for that job.

The setup times must be added to the normal durations of jobs in $D$, while the costs in $C$ must be subtracted from the profits in $P$.

The schedules for the 4 categories are given schematically in Figure 1. The values of the first objective function in each of the four categories are (in thousands) R128, R150, R124 and R124, respectively. In all the cases the optimum value of the second objective function is 12 time units.

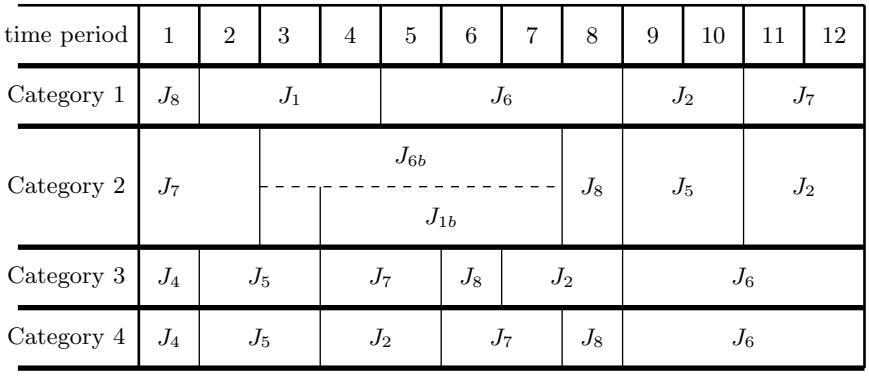


**Figure 1:** *A schematic representation of the optimal schedules for the four categories of scheduling problems.*

## 6 Conclusions and ideas for future research

Formulating a scheduling problem as an IP where job duration is dependent on the start time and the preceding job is relatively complex, but has the advantage of being very flexible. Many different scenarios may be modelled and additional constraints limiting job implementation times and precedence may easily be accommodated. Multiple objective functions may easily be incorporated by recursively running the same application with different objective functions and additional constraints imposed by enforcing previous objective values. The solution times of the IPP increases drastically as the pool of jobs and number of periods increase. The number of 0/1 variables and constraints become too large for commercial software to solve the IP model within an acceptable time frame. Fairly small but realistic examples of scenarios with 15 jobs and 30 time periods took days to converge to an optimal solution using LINGO. The possibilities for future research thus lies in the development of heuristics or meta-heuristics to find good job schedules as opposed to optimal job schedules.

## References

[1] Alidaee B & Womer NK, 1999, *Scheduling with time dependent processing times: Review and extentions*, Journal of the Operational Research Society, **50(7)**, pp. 711–720.

[2] Bachman A, Cheng TCE, Janiak A & Ng CT, 2002, *Scheduling start time dependent jobs to minimise the total weighted completion time*, Journal of the Operational Research Society, **53(6)**, pp. 688–693.

[3] Cheng TCE & Ding Q, 2001, *Single machine scheduling with step deteriorating processing times*, European Journal of Operational Research, **134(3)**, pp. 623–630.

[4] Cheng TCE & Ding Q, 2003, *Scheduling start time dependent tasks with deadlines and identical initial processing times on a single machine*, Computers and Operations Research, **30(1)**, pp. 51–62.

[5] Cheng TCE, Ding Q, Kovalyov MY, Bachman A & Janiak A, 2003, *Scheduling jobs with piecewise linearly decreasing processing times*, Naval Research Logistics, **50(6)**, pp. 531–554.

[6] Cheng TCE, Ding Q & Lin BMT, 2004, *A concise survey of scheduling with time-dependent processing times*, European Journal of Operational Research, **152(1)**, pp. 1–13.

[7] Gupta SK, Kunnather AS & Dandapani K, 1987, *Optimal repayment policies for multiple loans*, OMEGA, **15(4)**, pp. 323–330.

[8] Gupta JND & Gupta SK, 1988, *Single facility scheduling with nonlinear processing times*, Computers and Industrial Engineering, **14(4)**, pp. 387–393.

[9] Jeng AAK & Lin BMT, 2005, *Minimizing the total completion time in single-machine scheduling with step-deteriorating jobs*, Computers and Operations Research, **32(3)**, pp. 521–536.

[10] Kolman B & Beck RE, 1980, *Elementary linear programming with applications*, Academic Press Inc., Orlando (FL).

[11] Kubiak W & Van de Velde SL, 1998, *Scheduling deteriorating jobs to minimize makespan*, Naval Research Logistics, **45(5)**, pp. 511–523.

[12] LINGO, 2004, *Lingo: The modelling language and optimizer*, LINGO user's guide, LINDO Systems Inc., Chicargo (IL).

[13] LINGO, 2006, *Lindo systems' index page*, [Online], [Cited September 14th, 2006], Available from `http://www.lindo.com/`

[14] Pinedo M, 2002, *Scheduling theory, algorithms and systems*, $2^{nd}$ edition, Prentice-Hall Inc., Upper Saddle River (NJ).

[15] Zoints S, 1974, *Linear and integer programming*, Prentice-Hall Inc., Englewood Cliffs (NJ).