Sequencing spinning lines

Problem presented by

Jon Taylor

Acordis Acrylic Fibres, Grimsby

Problem statement

The Acordis acrylic fibres plant in Grimsby operates thirteen production lines, extruding four basic polymer types to make fibres. There are twelve key variables which define the end product. All changes to these variables take time (some more than others) and low grade product or waste is produced during the changeover. The most common product change is in the fibre colour. It is believed that optimised production scheduling can reduce the number or duration of the changes. Production scheduling is currently carried out by one skilled person who has to consider both production issues and customer requirements. The study group was asked to develop a tool to both help the scheduler and assist the production team leaders who periodically have to re-jig the production schedule outside of day hours and at short notice.

Study Group contributors

R. Eglese (University of Lancaster)
J. Gravesen (Technical University of Denmark)
P. G. Hjorth (Technical University of Denmark)
S. D. Howison (University of Oxford)
S. Khan (University of Cambridge)
R. A. Leese (Smith Institute)
A. N. Letchford (University of Lancaster)
S. D. Noble (Brunel University)
H. Tewkesbury (Smith Institute)

Report prepared by

Jens Gravesen (Technical University of Denmark)
Steven Noble (Brunel University)

1 Overview

The study group was asked by Acordis to assist in the production of a tool to schedule the production of acrylic fibres on a number of spinning lines. The group was informed by a presentation from Dr. Jon Taylor describing the acrylic fibre production process. Jobs are currently scheduled by a highly skilled and experienced scheduler without the use of any computational tool. Currently a schedule is produced each week and it takes several hours to produce the schedule.

In the next section we give a precise description of the problem, describing all the variables involved in the fibre production and the delays involved when a change is made to them. Many of the key variables may take only a small number of values and the delays involved in changing the settings associated with these variables are well known from experience. However a large range of colour dyes is used in the production of fibre and the potential range of colours that may be used in the future is unlimited. The amount of time needed to change between one colour and another is less well known and in Section 3 we present two suggestions for modelling it.

Section 4 describes an algorithm which, when given a set of jobs to be assigned to lines and sequenced, will produce a solution in a reasonable period of time. We close with conclusions, discussing the effectiveness and applicability of our proposed algorithm.

2 Background

Acordis produces a wide range of acrylic fibres, with specifications determined by twelve variables, including the polymer type, the thickness of the fibre and the colour. In detail, these variables and the values that they may take are as follows:

1. **Polymer.** The basic polymer type may take three values: standard (S), type ten (TX) and latent crimp (LC).
2. **Pigment.** The pigment may be one of three values: bright (B), matt (M) or duracoll black (DB).
3. **Stretch.** This refers to the tension placed on the fibres and may be high tension (HT), low tension (LT) or standard (-).
4. **Spin bath temperature.** This may be either standard or low.
5. **F setting.** This is a binary variable indicating whether the production process includes the characteristic known as a ‘relaxation’.
6. **Special pigments.** This variable indicates the use of a variety of special pigments and may take the values AB, BB, AF, BF, PCM or standard. However the AB and BB, and the AF and BF settings are really the same product for different manufacturers so essentially there are only 4 possible values for this variable.
7. **Decitex.** The decitex refers to the thickness of the individual fibres in the product and may take 12 possible values.
8. **Crimp.** There are 9 possibilities here.

9. **Soft finish.** This variable may take 5 possible values.
10. **Anti-static** Here there are two possibilities indicating whether an anti-static special finish is applied.
11. **Kilotex.** This is the total weight of fibre per unit length and is stated in grammes per metre.
12. **Colour.** This is the colour of the fibre.

Currently there are eight spinning lines in use, labelled by the letters A, B, C, D, G, H, J and K. Each line has different characteristics in that it may only produce certain types of fibre, according to the following table:

		A	B	C	D	G	H	J	K
Polymer	S	N	N	Y	Y	Y	Y	Y	Y
	TX	Y	Y	Y	Y	Y	Y	Y	Y
	LC	N	N	N	N	N	N	Y	Y
Pigment	B	Y	Y	Y	Y	Y	Y	Y	Y
	M	Y	Y	Y	Y	Y	Y	Y	Y
	DB	N	N	N	N	Y	N	N	Y
Stretch		All	All	All	All	All	All	All	All
Spin bath temp		Std	Std	Std	Std	All	All	All	All
F		N	N	N	N	N	Y	N	Y
Special pigment		N	N	N	N	N	Y	N	N
Decitex		> 2.2	> 2.2	> 2.2	> 2.2	All	All	All	All
Crimp		All	All	All	All	All	All	All	All
Soft finish		All	All	All	All	All	All	All	All
Anti-static		All	All	All	All	All	All	All	All
Kilotex		≤ 80	≤ 80	≤ 80	≤ 80	All	All	All	All
Colour		All	All	All	All	All	All	All	All

Table 1: Summary of possible fibres that may be produced on each spinning line

The entries in the table require some explanation. The first two rows list all possible values of the polymer and pigment type and indicate by means of a Y(es) or N(o) entry whether a product with a particular polymer and pigment type may be produced on a given line.

In the rest of the table the entry 'All' means that there is no restriction for the corresponding variable and line, an entry '> 2.2' or '≤ 80' indicates a range of possible values for a variable, and 'Std' indicates that only the standard setting for the variable can be produced. The row corresponding to the F setting indicates that products requiring the F setting can only be produced on lines H and K, and the row corresponding to special pigments show that these products can only be produced on line H.

In addition to these hard constraints there is a preference for products using the Duracoll Black pigment to be produced on line G.

Each of the lines consists of a number of different machines. Machines corresponding to two different lines do not necessarily run at the same speed. Consequently the eight spinning lines do not all produce fibre at the same speed. The following table shows the production speeds in metres per minute.

Line	A	B	C	D	G	H	J	K
Speed	264	264	264	308	190	190	190	190

Table 2: Speeds of fibre production (metres/sec) on each line

Changing the value of any of the variables causes a delay during which waste product of poor quality is produced. The delays depend not only on the variable being changed but also on the precise values of the variable involved in the change, and can range from 3 minutes to 2 hours. Some variable changes require certain key features of the line to be altered and production has to be temporarily stopped. This procedure is called a 'respin' and an important part of our proposed approach is to find a schedule that minimises the number of respins. Each respin takes 1 hour, during which no fibre is produced. More generally, we impose a penalty equal to the number of minutes of delay. The time taken in a change of variable, other than a colour change, is summarised in the following tables.

The first table is a matrix with (i, j)-entry representing the delay incurred when the polymer variable is changed from the ith possible value to the jth possible value. A respin is needed whenever the polymer is changed to or from latent crimp.

	S	TX	LC
S	0	30	60
TX	30	0	60
LC	60	60	0

Table 3: Delays incurred during polymer change (minutes)

The next table is a matrix with (i, j)-entry representing the delay incurred when the pigment variable is changed from the ith possible value to the jth possible value. A respin is never needed.

	B	M	DB
B	0	20	60
M	20	0	60
DB	120	120	0

Table 4: Delays incurred during pigment change (minutes)

For the remainder of the variables, with the exception of colour, the time taken for a change to the variable does not depend on the actual values involved. The delays, in minutes, that these changes incur are given below together with an indication (Y(es) / N(o)) of whether a respin is required.

Variable	Time	Re-Spin
Stretch	60	Y
Spin bath temp.	15	N
F	20	N
Special pigment	60	Y
Decitex	60	Y
Crimp	20	N
Soft finish	15	N
Anti-static	15	N
Kilotex	60	Y

Table 5: Delays incurred during assorted changes

The amount of time needed to for a change to the colour is much more complicated than any of the other variables. This is discussed in detail in the next section.

Each batch of fibre to be produced has a due date. The precise meaning of the due date is quite involved and depends on whether the product is to be delivered to a UK address or overseas. Most due dates correspond to Mondays. For a UK customer this means that the product should be despatched as early as possible within the week beginning on the due date whereas for an overseas customer the product must be despatched by the Friday preceding the due date. For our application we have chosen to simplify the treatment of due dates by assuming that each of them corresponds to a Friday by which the product must be despatched. For late jobs we impose a fixed penalty per tonne of product per week or part of a week that it is late.

The cost of a schedule is now given by adding the penalties due to delay and missed due dates. In the next section we discuss delays due to a colour change.

3 Colour space

In this section we will estimate the time for a colour change, using two different models.

The first we call the ‘dye model’. Here we assume that the old dye mixes with the new dye to produce a slightly wrong dye. We use the Kubelka-Munk method to model the dye process, and, assuming a small fraction of the old dye, a simple linearization yields the estimate.

The second approach is the ‘spot model’ where we assume that the old dye stains the new fabric in form of small spots of the old colour. If the spots are small, then the light reflected from the stained fabric will appear as a mix of light reflected from the old and the new fabric. Hence the colour will be a convex combination of the old and the new colour.

There are many ways to parameterize colour space, so in order to make our choice precise we have included some background material on colours, mostly taken from [11].

3.1 From light to colour

The human retina has three types of colour photorecepter cone cells, with different spectral sensitivities.[1] Therefore three real numbers are necessary and sufficient to describe a colour. We can think of the three numbers as the power received by each of the three different colour photorecepters.

The International Commission On Illumination (CIE[2]) has defined several possible parametrizations of the space of colours. If $S : [\lambda_1, \lambda_2] \to \mathbb{R}_+$ is the intensity function for the light, then the *CIE XYZ components* are defined by

$$(X, Y, Z) = k \int_{\lambda_1}^{\lambda_2} S(\lambda) \left(\overline{x}(\lambda), \overline{y}(\lambda), \overline{z}(\lambda)\right) \mathrm{d}\lambda,$$

where the functions $\overline{x}(\lambda)$, $\overline{y}(\lambda)$, and $\overline{z}(\lambda)$ are the CIE 1931 Standard Colorimetric Observers (see Figure 1) and k is a normalization constant which makes $Y = 100$ for a standard light source $S(\lambda)$, *i.e.*

$$k = \frac{100}{\int_{\lambda_1}^{\lambda_2} S(\lambda)\, \overline{y}(\lambda)\, \mathrm{d}\lambda}.$$

For the CIE standard illuminant D_{65} in Figure 1, we have $k = 0.047332$. The component Y is called the *luminance* and is an attempt to define the total observed intensity of the light.

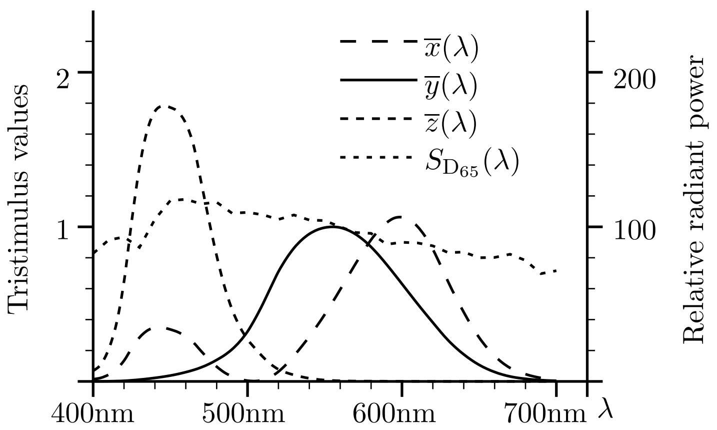

Figure 1: CIE 1931 Standard Colorimetric Observers and the spectral distribution for the CIE illuminant D_{65}. They are tabulated in [11] and can also be found at the CIE web-site.

The *CIE xy chromatic coordinates* are given by

$$x = \frac{X}{X + Y + Z}, \qquad y = \frac{Y}{X + Y + Z}.$$

[1]In fact a fourth type of photorecepter cells, the rod, is also present, but these are only used at extremely low light levels (night vision), and do not contribute to the perception of colour.

[2]http://members.eunet.at/cie/

Sometimes a third coordinate $z = Z/(X+Y+Z)$ is defined, but it can always be found from the relation $x+y+z=1$. The two chromatic coordinates x and y describe 'pure' colour, in the absence of luminance (or brightness). When monochromatic light sweeps over the visual light range from 400nm to 700nm, it traces a curve in the xy-space, shown in Figure 2. The line connecting the two ends of the curve is called the line of purples. It joins extreme blue with extreme red and consists consequently of mixtures of blue and red.

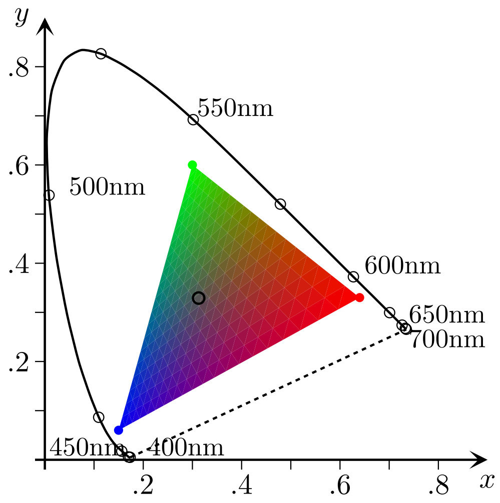

Figure 2: The tristimulus diagram. The monochromatic colours lie on the curved part of the boundary. The dashed line joining the the end of the visible spectrum [400nm, 700nm] is the line of purples. The triangle are the colours that can be produced by the primaries of the Rec. 709 RGB specifications, [1]. The circle indicates the D_{65} white point.

A colour can be specified by chromaticity (x, y) and luminance Y in the form of the *CIE xyY components*. To recover X and Z the following formulas are used:

$$X = Y\,\frac{x}{y}, \qquad Z = Y\,\frac{1-x-y}{y}.$$

The colours on a computer screen or a television are given by mixing three *primaries*: red, green, and blue. The three primaries correspond to three points in xy-space and the screen can reproduce all colours in the triangle spanned by the three primaries, the *gamut* of the primaries. In Figure 2 the primaries for the HDTV [1] are plotted and it is easily seen that not all colours can be obtained. The actual colours in the plot need not be correct, they depend on the computer screen, or on the printer and the illumination.

3.2 The colour of an object

The colour of an object depends not only on the object itself, but also on the illumination. The influence from the object is given by the *reflection coefficient* $\beta(\lambda)$, which specifies how light is reflected at different wavelengths. If the illumination has the spectral distribution $S(\lambda)$, then the object emits light with the spectral distribution $\beta(\lambda)\, S(\lambda)$. Hence the CIE XYZ components are given by

$$(X, Y, Z) = k \int_{\lambda_1}^{\lambda_2} \beta(\lambda)\, S(\lambda) \left(\overline{x}(\lambda), \overline{y}(\lambda), \overline{z}(\lambda)\right) \mathrm{d}\lambda.$$

We obviously have $0 \leq \beta(\lambda) \leq 1$ so, if the illumination is fixed, then any object can at most emit light with intensity $S(\lambda)$. So the set of possible colours in a fixed illumination is a bounded set; see [11, Figure 2(3.3.9)] and Figure 4.

3.3 Colour differences

The human perception of similar colours has little to do with the Euclidean distance in the xy-plane. Indeed, some sixty years ago MacAdam conducted some colour matching experiments where a person was asked to match a colour with given chromatic coordinates (x, y) by adjusting another colour by a single control that traced a line through (x, y) in the chromatic plane. The standard deviations turned out to be ellipses in the chromatic plane, sketched in Figure 3.

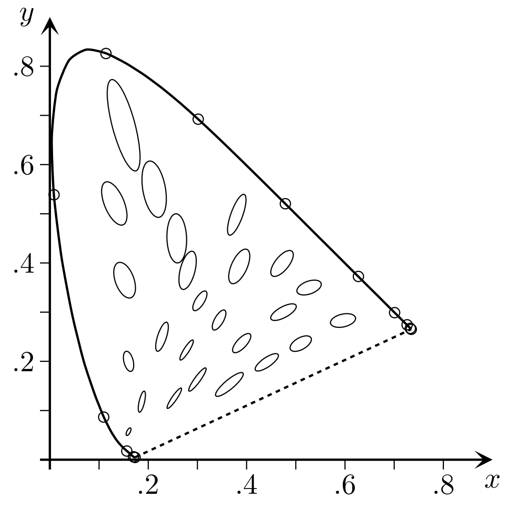

Figure 3: The MacAdam ellipses [7], enlarged 10 times. If the depicted ellipses are diminished by a factor of three then colours on the ellipse can just be seen to be different from the colour at the centre. The parameters of the ellipses are tabulated in [11].

If the ellipses are enlarged approximately three times they define the *just noticeable difference*, *i.e.* colours inside the enlarged ellipse appear to be the same as the one at the centre while colours outside appear to be different from the colour at the centre. This discrepancy between human perception and the Euclidean distance has spawned attempts to define parameters which are more uniform with respect to the human perception. The standard we use is the *CIELAB coordinates* which are given by

$$L^* = \begin{cases} 903.3\, Y/Y_n & \text{if } Y/Y_n \le 0.008856 \\ 116\sqrt[3]{Y/Y_n} - 16 & \text{if } Y/Y_n > 0.008856 \end{cases}$$

$$\begin{aligned} a^* &= 500\big(f(X/X_n) - f(Y/Y_n)\big) \\ b^* &= 200\big(f(Y/Y_n) - f(Z/Z_n)\big) \end{aligned}$$

where

$$f(t) = \begin{cases} 7.787\, t + 16/116 & \text{if } t \le 0.008856 \\ \sqrt[3]{t} & \text{if } t > 0.008856. \end{cases}$$

The inverse map is

$$\frac{Y}{Y_n} = \begin{cases} L^*/903.3 & \text{if } L^* \le 8 \\ \big((L^* + 16)/116\big)^3 & \text{if } L^* > 8 \end{cases} \tag{1}$$

$$\frac{X}{X_n} = f^{-1}\big(a^*/500 - f(Y/Y_n)\big) \tag{2}$$

$$\frac{Z}{Z_n} = f^{-1}\big(f(Y/Y_n) - b^*/200\big) \tag{3}$$

where

$$f^{-1}(t) = \begin{cases} (t - 16/116)/7.787 & \text{if } t \le 0.2069 \\ t^3 & \text{if } t > 0.2069. \end{cases}$$

The triple (X_n, Y_n, Z_n) are the components of the white reference, where Y_n is normalized to 100. For the D_{65} white point we have the values $(X_n, Y_n, Z_n) = (95.043, 100, 108.88)$. In the *CIELCH coordinates* the Cartesian coordinates (a^*, b^*) are replaced by polar coordinates (C, H) called *chroma* and *hue* respectively,

$$a^* = C \cos H, \qquad b^* = C \sin H.$$

As we can see in Figure 4 the size of the ellipses has become somewhat more uniform, but they are still far from circles (of equal size).

We will now describe the CMC[3] colour difference formula, but first we need some definitions. Let $L_1^*, a_1^*, b_1^*, C_1^*, H_1^*$ and $L_2^*, a_2^*, b_2^*, C_2, H_2$ be the coordinates of two colours. Then we let

$$\Delta L^* = L_1^* - L_2^*, \qquad \Delta a^* = a_1^* - a_2^*, \qquad \Delta b^* = b_1^* - b_2^*.$$

Likewise $\Delta C = C_1 - C_2$, but ΔH is defined such that the equation $(\Delta C)^2 + (\Delta H)^2 = (\Delta a^*)^2 + (\Delta b^*)^2$ holds, *i.e.*

[3]The Colour Measurement Committee of the Society of Dyers and Colourists, whose web site may be found at http://www.sdc.org.uk/

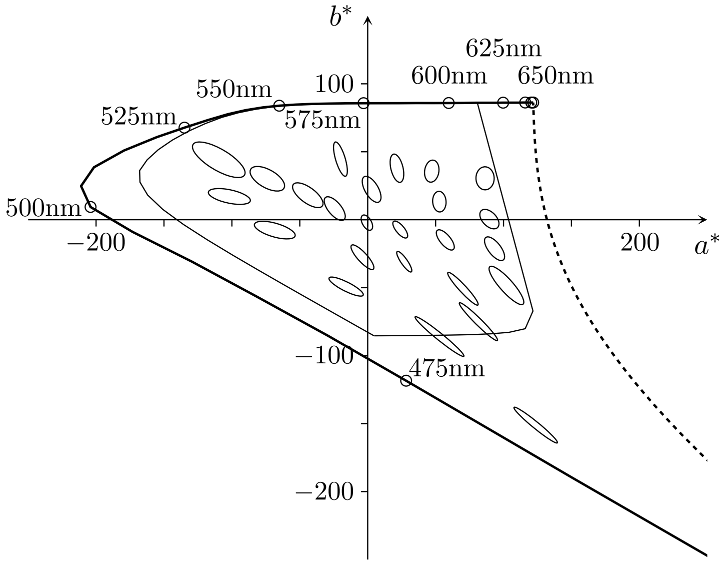

Figure 4: The MacAdam ellipses in the a^*b^*-plane, $L^* = 50$. The smaller region contains the possible colours of an object illuminated by the CIE D_{65} standard.

$$\begin{aligned}(\Delta H)^2 &= (\Delta a^*)^2 + (\Delta b^*)^2 - (\Delta C)^2 \\ &= (C_1 \cos H_1 - C_2 \cos H_2)^2 + (C_1 \sin H_1 - C_2 \sin H_2)^2 - (C_1 - C_2)^2 \\ &= 2C_1\, C_2 (1 - \cos H_1\, \cos H_2 - \sin H_1\, \sin H_2) \\ &= 2C_1\, C_2 (1 - \cos(H_1 - H_2)) \\ &= 4C_1\, C_2\, \sin^2\left(\frac{H_1 - H_2}{2}\right).\end{aligned}$$

The CMC colour difference formula allows calculation of tolerance ellipsoids around a given colour where the dimensions of the ellipsoid are a function of the given colour. The design of this formula allows for two user-definable coefficients ℓ and c and the formula is thus normally specified as CMC(ℓ:c). The values of ℓ and c modify the relative importance that is given to differences in lightness and chroma respectively. The CMC(2:1) version of the formula has been shown to be useful for the estimation of the acceptability of colour difference evaluations.

The CMC(2:1) equation is a British Standard (BS:6923) for the assessment of small colour differences and is currently being considered as an ISO standard. The CMC(ℓ:c)-distance from colour 1 to colour 2 is defined as

$$\Delta E = \sqrt{\left(\frac{\Delta L^*}{\ell S_L(L_1^*)}\right)^2 + \left(\frac{\Delta C}{c S_C(C_1)}\right)^2 + \left(\frac{\Delta H}{S_H(C_1, H_1)}\right)^2},$$

where

$$S_L(L) = \begin{cases} \dfrac{0.040975L}{1+0.01765L} & \text{if } L \geq 16 \\ 0.511 & \text{if } L < 16 \end{cases} \tag{4}$$

$$S_C(C) = \frac{0.0638C}{1+0.0131C} + 0.638,$$

$$S_H(C,H) = S_C(C)\big(1 - f(C) + f(C)T(H)\big),$$

with

$$f(C) = \frac{C^2}{\sqrt{C^4 + 1900}},$$

$$T(H) = \begin{cases} 0.36 + |0.4\cos(H+35)| & \text{if } -15 \leq H \leq 164 \\ 0.56 + |0.2\cos(H+168)| & \text{if } 164 < H < 345 \end{cases}$$

and H is measured in degrees. Notice that this notion of distance is asymmetric. The distance from colour 1 to colour 2 is most likely not the same as the distance from colour 2 to colour 1. For small differences we have

$$\Delta C \approx \frac{a^*\Delta a^* + b^*\Delta b^*}{\sqrt{a^{*2} + b^{*2}}} \quad \text{and} \quad \Delta H \approx \frac{-b^*\Delta a^* + a^*\Delta b^*}{\sqrt{a^{*2} + b^{*2}}}.$$

Hence

$$\Delta C^2 \approx \frac{a^{*2}\Delta a^{*2} + 2a^*b^*\Delta a^*\Delta b^* + b^{*2}\Delta b^{*2}}{a^{*2} + b^{*2}},$$

$$\Delta H^2 \approx \frac{b^{*2}\Delta a^{*2} - 2a^*b^*\Delta a^*\Delta b^* + a^{*2}\Delta b^{*2}}{a^{*2} + b^{*2}},$$

and in CIELAB coordinates we have

$$\Delta E^2 = G_L(\Delta L^*)^2 + G_a(\Delta a^*)^2 + G_{ab}\Delta a^*\Delta b^* + G_b(\Delta b^*)^2\,, \tag{5}$$

where

$$G_L(L_1^*) = (\ell S_L(L_1^*))^{-2} \tag{6}$$

$$G_a(a,b) = \frac{S_2(a,b)^2a^2 + c^2b^2}{(a^2+b^2)c^2S_1(a,b)^2S_2(a,b)^2}, \tag{7}$$

$$G_{ab}(a,b) = \frac{2ab(S_2(a,b)^2 - c^2)}{(a^2+b^2)c^2S_1(a,b)^2S_2(a,b)^2}, \tag{8}$$

$$G_b(a,b) = \frac{c^2a^2 + S_2(a,b)^2b^2}{(a^2+b^2)c^2S_1(a,b)^2S_2(a,b)^2}, \tag{9}$$

and

$$S_1(a,b) = \frac{0.0638\sqrt{a^2+b^2}}{1+0.0131\sqrt{a^2+b^2}} + 0.638, \tag{10}$$

$$S_2(a,b) = 1 - g(a,b) + g(a,b)h(a,b), \tag{11}$$

$$g(a,b) = \frac{a^2+b^2}{\sqrt{(a^2+b^2)^2+1900}}, \tag{12}$$

$$h(a,b) = \begin{cases} 0.36 + \dfrac{|0.3277\,a - 0.2294\,b|}{\sqrt{a^2+b^2}} & \text{if } 0.2588\,a + 0.9659\,b \geq 0, \\ 0.56 + \dfrac{|0.1956\,a + 0.04158\,b|}{\sqrt{a^2+b^2}} & \text{if } 0.2588\,a + 0.9659\,b < 0. \end{cases} \tag{13}$$

Similarly

$$\Delta L^* \approx L^{*\prime}\left(\frac{Y_2}{Y_n}\right)\frac{\Delta Y}{Y_n} \tag{14}$$

$$\Delta a^* \approx 500\left(f'\left(\frac{X_2}{X_n}\right)\frac{\Delta X}{X_n} - f'\left(\frac{Y_2}{Y_n}\right)\frac{\Delta Y}{Y_n}\right) \tag{15}$$

$$\Delta b^* \approx 200\left(f'\left(\frac{Y_2}{Y_n}\right)\frac{\Delta Y}{Y_n} - f'\left(\frac{Z_2}{Z_n}\right)\frac{\Delta Z}{Z_n}\right), \tag{16}$$

where

$$L^{*\prime}(t) = \begin{cases} 903.3 & \text{for } t \leq 0.008856 \\ \dfrac{116}{3}t^{-2/3} & \text{for } t > 0.008856 \end{cases} \tag{17}$$

$$f'(t) = \begin{cases} 7.787 & \text{for } t \leq 0.008856 \\ \dfrac{1}{3}t^{-2/3} & \text{for } t > 0.008856. \end{cases} \tag{18}$$

Thus, in CIEXYZ coordinates the CMC-equation becomes

$$\Delta E^2 = G_X\left(\frac{\Delta X}{X_n}\right)^2 + G_Y\left(\frac{\Delta Y}{Y_n}\right)^2 + G_Z\left(\frac{\Delta Z}{Z_n}\right)^2 + G_{XY}\frac{\Delta X}{X_n}\frac{\Delta Y}{Y_n} + G_{YZ}\frac{\Delta Y}{Y_n}\frac{\Delta Z}{Z_n} + G_{XZ}\frac{\Delta X}{X_n}\frac{\Delta Z}{Z_n}\ , \tag{19}$$

where

$$G_X = 500^2 G_a f' \left(\frac{X_1}{X_n}\right)^2 \tag{20}$$

$$G_Y = G_L L^{*\prime} \left(\frac{Y_1}{Y_n}\right)^2 + \left(500^2 G_a - 500 \cdot 200\, G_{ab} + 200^2 G_b\right) f' \left(\frac{Y_1}{Y_n}\right)^2 \tag{21}$$

$$G_Z = 200^2 G_b f' \left(\frac{Z_1}{Z_n}\right)^2 \tag{22}$$

$$G_{XY} = \left(500 \cdot 200\, G_{ab} - 2 \cdot 500^2 G_a\right) f' \left(\frac{X_1}{X_n}\right) f' \left(\frac{Y_1}{Y_n}\right) \tag{23}$$

$$G_{YZ} = \left(500 \cdot 200\, G_{ab} - 2 \cdot 200^2 G_b\right) f' \left(\frac{Y_1}{Y_n}\right) f' \left(\frac{Z_1}{Z_n}\right) \tag{24}$$

$$G_{XZ} = -500 \cdot 200\, G_{ab} f' \left(\frac{X_1}{X_n}\right) f' \left(\frac{Z_1}{Z_n}\right) \tag{25}$$

The MacAdam ellipses have roughly 1/3 of the size of the CMC-tolerance ellipses, as they should, but the shapes of the two sets of ellipses are quite different; see Figure 5. This need not concern us here, as the goal of the dye process is to be within the CMC-tolerance, or perhaps the Marks & Spencer[4] tolerance.

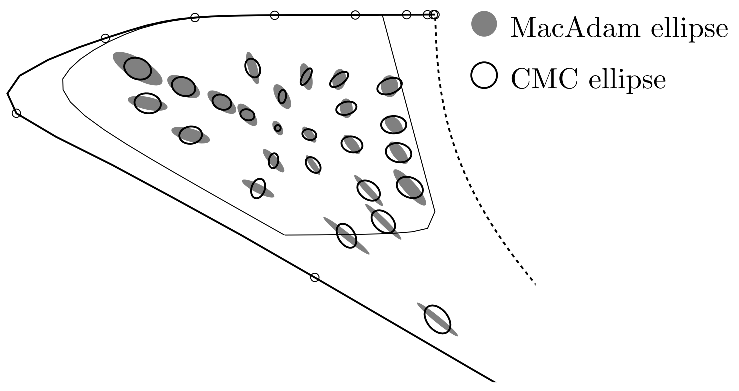

Figure 5: The MacAdam ellipses and the CMC-ellipses in the a^*b^*-plane, $L^* = 50$.

3.4 The dye model

Just as in [3] we will use the Kubelka-Munk model — see also [8] — for the dye process. If a material is dyed with a mixture of N dyes, each at a concentration C_n, then the

[4]In the 1980s Marks & Spencer, in conjunction with Instrumental Colour Systems, developed their own in-house equations that are used in the textile industry. The M&S equations have never been published, but are reported to give similar results as the CMC equations. In any case it should not be difficult to replace the CMC-standard with the M&S-standard.

reflection coefficient is given by

$$\beta(\lambda) = 1 + \frac{k(\lambda)}{s(\lambda)} - \sqrt{\left(1 + \frac{k(\lambda)}{s(\lambda)}\right)^2 - 1}.$$

Conversely $k/s = (1-\beta)^2/(2\beta)$, and $\sqrt{(1+k/s)^2 - 1} = (1-\beta^2)/(2\beta)$. In terms of the dye concentrations, the fraction k/s can be calculated from

$$\frac{k(\lambda)}{s(\lambda)} = \frac{k_0(\lambda)}{s(\lambda)} + \sum_{n=1}^{N} \frac{k_n(\lambda)}{s(\lambda)} \frac{C_n}{C_0},$$

where $k_n(\lambda)$ is the absorption of dye n at concentration C_0, and k_0 is the absorption of the undyed material. The equation basically says that all the scattering, $s(\lambda)$, is due to the undyed material, while the absorption, $k(\lambda)$, may be superimposed linearly.

Now assume we have two dye recipes given by concentrations C_n^1 and C_n^2 respectively. When the process shifts from C_n^1 to C_n^2, a small fraction εC_n^1 of the old dye is left and we will assume that it decays exponentially, as $\varepsilon e^{-\mu t} C_n^1$. We then have

$$\frac{k(\lambda)}{s(\lambda)} = \underbrace{\frac{k_0(\lambda)}{s(\lambda)} + \sum_{n=1}^{N} \frac{k_n(\lambda)}{s(\lambda)} \frac{C_n^2}{C_0}}_{k_2(\lambda)/s(\lambda)} + \varepsilon e^{-\mu t} \underbrace{\sum_{n=1}^{N} \frac{k_n(\lambda)}{s(\lambda)} \frac{C_n^1}{C_0}}_{\widehat{k}_1(\lambda)/s(\lambda)}$$

If $(2(1+k_2/s) + \widehat{k}_1/s\, \varepsilon e^{-\mu t})\, \widehat{k}_1/s\, \varepsilon e^{-\mu t}$ is small compared to $(1+k_2/s)^2 - 1$, we have

$$\begin{aligned}
&\sqrt{\left(1 + \frac{k_2}{s} + \frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t}\right)^2 - 1} \\
&\qquad = \sqrt{\left(1 + \frac{k_2}{s}\right)^2 - 1 + 2\left(1 + \frac{k_2}{s}\right)\frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t} + \left(\frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t}\right)^2} \\
&\qquad \approx \sqrt{\left(1 + \frac{k_2}{s}\right)^2 - 1}\left(1 + \frac{1 + k_2/s}{(1+k_2/s)^2 - 1}\frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t}\right) \\
&\qquad = \sqrt{\left(1 + \frac{k_2}{s}\right)^2 - 1} + \frac{1 + k_2/s}{\sqrt{(1+k_2/s)^2 - 1}}\frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t}.
\end{aligned}$$

So if $\beta_2(\lambda)$ is the reflection coefficient for the second colour, then the reflection coefficient of the mixed colour is

$$\begin{aligned}
\beta(\lambda) &= 1 + \frac{k_2}{s} + \frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t} - \sqrt{\left(1 + \frac{k_2}{s} + \frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t}\right)^2 - 1} \\
&\approx 1 + \frac{k_2}{s} - \sqrt{\left(1 + \frac{k_2}{s}\right)^2 - 1} + \left(1 - \frac{1 + k_2/s}{\sqrt{(1+k_2/s)^2 - 1}}\right)\frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t} \\
&= \beta_2 - \frac{\beta_2}{(1-\beta_2^2)/(2\beta_2)}\frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t} = \beta_2 - \frac{2\beta_2^2}{1-\beta_2^2}\frac{\widehat{k}_1}{s}\varepsilon e^{-\mu t}.
\end{aligned}$$

If $(\widehat{k}_1/s)^2\,\varepsilon^2 \ll 2\beta_2^2/(1-\beta_2^2)\,(\widehat{k}_1/s)\,\varepsilon \ll 1$, this is a good approximation. The dye concentrations C_n/C_0, the absorption coefficients k_n/s, and the reflection coefficient β_2 are all quantities known to Acordis, so we only need to determine ε and μ. At this point we can already see that if the second colour is lighter, then β_2 is closer to 1, and the initial error term $2\beta_2^2/(1-\beta_2^2)\,(\widehat{k}_1/s)\,\varepsilon$ is larger, thus it will take longer before the term is negligible.

The CIE XYZ components are given by

$$\begin{aligned}
(X,Y,Z) &= k\int \beta(\lambda)\,S(\lambda)\,\big(\overline{x}(\lambda),\overline{y}(\lambda),\overline{z}(\lambda)\big)\,\mathrm{d}\lambda \\
&= k\int\left(\beta_2(\lambda) - \frac{2\beta_2(\lambda)^2}{1-\beta_2(\lambda)^2}\,\frac{\widehat{k}_1(\lambda)}{s(\lambda)}\,\varepsilon e^{-\mu t}\right) S(\lambda)\,\big(\overline{x}(\lambda),\overline{y}(\lambda),\overline{z}(\lambda)\big)\,\mathrm{d}\lambda \\
&= k\int \beta_2(\lambda)\,S(\lambda)\,\big(\overline{x}(\lambda),\overline{y}(\lambda),\overline{z}(\lambda)\big)\,\mathrm{d}\lambda \\
&\quad - \varepsilon e^{-\mu t}\,2k\int \frac{2\beta_2(\lambda)^2}{1-\beta_2(\lambda)^2}\,\frac{\widehat{k}_1(\lambda)}{s}\,S(\lambda)\,\big(\overline{x}(\lambda),\overline{y}(\lambda),\overline{z}(\lambda)\big)\,\mathrm{d}\lambda.
\end{aligned}$$

If (X_2,Y_2,Z_2) are the components for colour 2, then we have

$$X = X_2 - \varepsilon\, e^{-\mu t}\Delta X,$$

with similar expressions for the Y and Z components, where

$$(\Delta X,\Delta Y,\Delta Z) = 2k\int \frac{2\beta_2(\lambda)^2}{1-\beta_2(\lambda)^2}\,\frac{\widehat{k}_1(\lambda)}{s}\,S(\lambda)\,\big(\overline{x}(\lambda),\overline{y}(\lambda),\overline{z}(\lambda)\big)\,\mathrm{d}\lambda, \tag{26}$$

Note that we can write

$$\frac{\widehat{k}_1(\lambda)}{s} = \frac{(1-\beta_1(\lambda))^2}{2\beta_1(\lambda)} - \frac{(1-\beta_0(\lambda))^2}{2\beta_0(\lambda)}$$

where β_0 and β_1 are the reflection coefficients for the undyed material and the material dyed with colour 1, respectively. If the initial errors $\varepsilon\,\Delta X$, $\varepsilon\,\Delta Y$, and $\varepsilon\,\Delta Z$ are small then we can write the CMC-error as $\Delta E = \Delta E_0\,\varepsilon\, e^{-\mu t}$, where ΔE_0 is given by (19). The time for a colour change is given by the equation $\Delta E = 1$, *i.e.*

$$t = \frac{\ln(\Delta E_0^2) + 2\ln\varepsilon}{2\mu}. \tag{27}$$

The quantity ΔE_0 can be calculated with information available from Acordis. By looking back at old production data or by conducting controlled experiments it is possible to plot t as a function of $\ln(\Delta E_0)$. If the assumptions we have made here hold, then the data should fit well to a straight line, except for small values of ΔE_0. There is a minimum time for a colour change, so the graph should consist of a horizontal line followed by another straight line. The slope of the latter is $1/\mu$ and its intersection with the axis $\ln(\Delta E_0) = 0$ is $\ln\varepsilon/\mu$.

These values of μ and ε can now be used to predict the time of any colour change. All the above relies on the assumption that we can consider the tainting of the new colour with the old colour as single dye process. This need not be true. For example, the tainting could be due to floss of old fabric attaching itself to the new fabric, and it is not clear what the best model for this would be. In any case the procedure outlined above will tell us how good the model is.

3.5 The spot model

Instead of considering the residues of the old dye as a 'correction' to the new dye, we could adopt the following point of view. We have some residues of the old dye or colour scattered along the production line and they produce small spots of the old colour on the new fabric. If the individual spots are too small to be seen, then the effect of, say blue spots on a yellow fabric, would be to make the yellow colour more greenish. In the CIEXYZ space we would have:

$$\begin{aligned}(X, Y, Z) &= (1-\alpha)(X_2, Y_2, Z_2) + \alpha(X_1, Y_1, Z_1) \\ &= (X_2, Y_2, Z_2) + \alpha(\Delta X, \Delta Y, \Delta Z),\end{aligned}$$

where (X_2, Y_2, Z_2) and (X_1, Y_1, Z_1) are the components of the new and old colour respectively, $\Delta X = X_1 - X_2$ and similar for Y and Z, and α is the fraction of the old colour present on the fabric. If we once more assume exponential decay of the old colour then we have

$$(X, Y, Z) = (X_2, Y_2, Z_2) + \varepsilon\, e^{-\mu t}\,(\Delta X, \Delta Y, \Delta Z). \tag{28}$$

Once more we can write the CMC-error as $\Delta E = \Delta E_0\, \varepsilon\, e^{-\mu t}$, where ΔE_0 is found by substituting (28) into (19). Consequently (27) is the time for a colour change, and we can calibrate and check this model in exactly the same manner as in the previous section. The only difference is the way we calculate ΔE_0. The present model has the advantage of being a function of the two colours only; we do not need the specific dye recipes.

If we only want to compare colour changes then we do not need the values of ε and μ, because $1/\mu$ is just a common factor and ε/μ is added to every colour change time. But if we want to compare colour changes with other kinds of changes in the production, then we do need these values.

At the study group we had the CIELCH values for seven different colours, given in Table 6.

Colour no.	1	2	3	4	5	6	7
L^*	59.40	44.60	37.16	83.75	79.01	28.22	84.48
C	62.94	11.64	50.29	81.47	16.10	19.99	8.39
H	162.86	148.27	166.69	139.60	168.05	85.00	175.83

Table 6: The CIELCH components of the seven colours.

If the 'colour spot' model is valid, then the colour change scheduling is determined by the logarithm of the CMC(1:1)-distances in CIEXYZ-space. They can be determined by the following algorithm. For each colour (L_i^*, C_i, H_i) we calculate:

$$\begin{aligned}
(a_i^*, b_i^*) \quad & = (C_i \cos H_i, C_i \sin H_i), \\
g(a_i^*, b_i^*) \text{ and } h(a_i^*, b_i^*) \quad & \text{using (12) and (13)}, \\
S_1(a_i^*, b_i^*) \text{ and } S_2(a_i^*, b_i^*) \quad & \text{using (10) and (11)}, \\
G_L(L^*),\ G_a(a_i^*, b_i^*),\ G_{ab}(a_i^*, b_i^*), \text{ and } G_b(a_i^*, b_i^*) \quad & \text{using (6), (7), (8) and (9)}, \\
G_X,\ G_Y,\ G_Z,\ G_{XY},\ G_{YZ}, \text{ and } G_{XZ} \quad & \text{using (20) – (25)}, \\
Y_i/Y_n,\ X_i/X_n, \text{ and } Z_i/Z_n \quad & \text{using (1), (2) and (3)}, \\
L^{*\prime}(Y_i/Y_n) \quad & \text{using (17)}, \\
f'(Y_i/Y_n),\ f'(X_i/X_n), \text{ and } f'(Z_i/Z_n) \quad & \text{using (18)}.
\end{aligned}$$

For each $j = 1, \ldots, 7$ we then calculate

$$\begin{aligned}
\Delta X_{ij}/X_n &= X_j/X_n - X_i/X_n, \\
\Delta Y_{ij}/Y_n &= Y_j/Y_n - Y_i/Y_n, \\
\Delta Z_{ij}/Z_n &= Z_j/Z_n - Z_i/Z_n, \\
\Delta E_{ij}^2 & \quad \text{using (19)}.
\end{aligned}$$

The result is the following distance matrix:

$$\left[\ln(\Delta E_{ij}^2)/2\right] = \begin{bmatrix} — & 5.10 & 4.64 & 4.22 & 4.34 & 6.19 & 4.84 \\ 4.11 & — & 2.71 & 4.79 & 5.17 & 5.29 & 5.54 \\ 4.19 & 2.98 & — & 4.80 & 5.19 & 4.99 & 5.55 \\ 4.55 & 6.08 & 5.62 & — & 4.19 & 6.98 & 4.07 \\ 4.09 & 5.85 & 5.39 & 3.43 & — & 6.78 & 3.55 \\ 4.64 & 4.76 & 3.89 & 5.05 & 5.51 & — & 5.84 \\ 4.41 & 6.03 & 5.57 & 2.96 & 3.37 & 6.93 & — \end{bmatrix}. \tag{29}$$

In the i'th row and j'th column we have the time (up to the parameters ε and μ) it takes to change from the i'th colour to the j'th colour.

In both the present model and the dye model we may replace the exponential decay $e^{-\mu t}$ with some other form of decay, say $\varrho(t)$. We then have the equation $\Delta E = \Delta E_0\, \varepsilon\, \varrho(t)$ for the CMC-difference, and (27) is replaced by $t = \varrho^{-1}\big(1/(\Delta E_0\, \varepsilon)\big)$.

4 Solution method

We now describe an algorithm to find a good schedule. To begin with we restrict attention to the case where we are only concerned with a single week's worth of jobs, with all lines free at the beginning of the week. In the final section we describe how to amend the procedure to deal with a rolling schedule. Some of the jobs will have a due date corresponding to the end of a week, some will have a due date further in the future and some will already be late.

The key to this procedure is to split the set of jobs into blocks, where a block is a maximal set of jobs for which the set-up time to change from one job to another is strictly less than thirty minutes, and which may only contain jobs of a single special pigment type. Notice that if a respin is required to change the settings from J_1 to J_2 then J_1 and J_2 are contained in separate blocks, however the converse in not true. In the following we will loosely refer to a respin occuring between any two blocks since it is only the time involved in changing the configuration of a line between jobs that concerns us and not the precise procedure. Clearly the first step in any procedure should be to minimise the number of lengthy delays due to respins, and partitioning jobs into blocks is the first stage in achieving this. Blocks containing only jobs with due dates later than the end of the week are now discarded as they may be scheduled at a later stage.

4.1 Assigning blocks to lines

Suppose we have blocks $B_1, B_2, \ldots, B_n$. We now describe the first step of the scheduling procedure which assigns blocks to lines. This is done using a version of depth first search (DFS), assigning blocks one by one until either it is impossible to assign more blocks or an assignment of all blocks is reached. Let n_i denote the number of jobs in block B_i, l_i denote the total length in km of fibre to be produced in B_i and let s_i denote the minimum set-up time required during the processing of block B_i. For each i we estimate s_i using the farthest insertion heuristic described in the next section. We label the lines $L_1, \ldots, L_8$ in some way and let r_j denote the length in km of fibre produced by line L_j in one hour (assuming no delays). Thus an estimate of the number of hours needed to process block B_i on line L_j is $l_i/r_j + s_i$ plus the amount of time needed for a respin at the start of the block. For each block B_i we let F_i denote the set of lines on which block B_i may be processed.

Throughout the process we measure the capacity available on each line in terms of time. The *actual capacity* a_j is the amount of free time available on line L_j until the end of the week, and the *useful capacity* u_j is a more complicated measure which allows some jobs to run late.

In order to determine the useful capacities to allocate to each line we next make an estimate of the total time, in machine-hours, required to process all of the blocks. This is given by

$$\hat{T} = \sum_{i=1}^{n} \left(s_i + \frac{8l_i}{M} + \left\lfloor \frac{l_i}{21M} \right\rfloor \right) + n - 8,$$

where $M = \sum_{j=1}^{8} r_j$, the total length of fibre produced per hour across all lines. The terms in the estimate correspond respectively to the delays due to changing settings within a block, the time to actually produce the fibre, time spent in respins because a large block must be split between lines[5] and other time spent in respins. Now let $T = \max\{\hat{T}, 1344\}$ (note that $1344 = 168 \times 8$ is the total number of machine-hours in a week). The time T is then split up equally between the lines and an extra amount of time added to allow for the possibility of avoiding some respins by allowing jobs to run

[5]Note that $8l_i/M$ is the time in hours taken to produce the fibre and dividing this by 168 gives the number of respins needed because the block exceeds the time available in a single week

late. More precisely we define u_j by

$$u_j = T/8 + \frac{1}{0.066\alpha r_j},$$

where α is the penalty per tonne of product for each week or part week that it is late. The second term comes from choosing to add on a length of time which is chosen so that if all jobs in that period are late then the ensuing cost would equal the cost of an extra respin. It is based on an average kilotex of 66 grammes/metre. The values of a_j and u_j are constantly updated in the procedure as they are used to represent the available capacity given the current partial assignment.

The blocks are ordered for consideration by the DFS in the following way. Firstly they are split into two types: those with total length greater than 100km or which may be processed on four or fewer lines, and the remaining blocks. The blocks of the first type are assigned by the DFS before those of the second type. Within each type, blocks are ordered firstly by the size of the set of lines F_i on which they may be scheduled and secondly by the total length of the block.

During the DFS, blocks are assigned one by one to a line in the order described above. Block B_i may be assigned to line L_j if $j \in F_i$ and the useful capacity remaining on line L_j is at least $s_i + l_i/r_j$. Each time a block is assigned to a line L_j the useful and actual capacities of line L_j are updated to take into account the time processing the block including the delays from changing settings within the block and time involved in a respin before a new block can be processed afterwards. Should a block be reached with, for every j, $s_i + l_i/r_j > u_j$ this block is not assigned at this stage but placed at the end of the blocks of the first type, to be split up between lines, as described below. If at any stage it becomes impossible to allocate a block because all the lines on which it may be scheduled have used up their useful capacity, then that branch of the DFS is abandoned and backtracking occurs.

Splitting large blocks between lines. When the DFS has constructed a partial assignment of all those blocks of the first type which do not require splitting it moves on to consider how to split the large blocks left over. Typically there will only be one or two blocks requiring splitting and these blocks are considered in all possible orders by the DFS. Once the DFS arrives at this stage no more backtracking occurs. A given block is split up using the following greedy heuristic, which ignores the fact that a block consists of discrete jobs and assigns a proportion of each block to particular lines.

The proportion π_{ij} of block B_i that may be assigned to line L_j (assuming that $j \in F_i$) without exceeding the useful capacity of L_j is

$$\frac{r_j u_j}{s_i r_j + l_i}.$$

To split a block B_i between lines, those lines in F_i are ordered in decreasing order of π_{ij}. Proportions of B_i are assigned to each of these lines in order, filling up the useful capacity, until there is a line with enough useful capacity to take the remaining proportion of the block. Once this happens we try to assign the last proportion of B_i to the line with the smallest *actual* capacity for the last proportion of B_i to fit. If no such line exists we assign the last proportion of B_i to the line with the smallest *useful* capacity large enough to take the final proportion of B_i. In the case where the final proportion of B_i does not use up all the actual capacity of the line L_l to which it is assigned, we allocate as many jobs from each of the other proportions of B_i in turn until the actual capacity of L_l is used up. This procedure is repeated with all the other blocks that have been split.

Where possible, the blocks on each line are now ordered in decreasing order of $\bar{W}_i q_i$ where $\bar{W}_i$ is the average weight (in tonnes) of a job in block B_i and q_i is the proportion of jobs in block B_i that have due date at the end of the week or before. When there is more than one block involving the production of jobs with a special finish this may not be possible.

We can estimate the cost C_j of late jobs on line j as follows. If a block B_i is started after the end of the week, then all jobs in the block with due dates at the end of the week or before are late and so the estimated cost of these is $\alpha \bar{W}_i q_i$. If block B_i is started before the end of the week but finishes after the end of the week and we assume that late jobs are equally likely to occur at any position in B_i then if T_f and T_s are estimates of the finishing and starting times (measured from the start of the week) of the block, respectively, an estimate of the cost of late jobs is

$$\alpha \left\lceil \frac{T_f - 168}{T_f - T_s} n_i \right\rceil q_i \bar{W}_i.$$

The small blocks of type two are now assigned greedily, each one being assigned to the line and in the position causing the smallest increment in total cost as a result of late jobs.

Possible reallocation of split blocks. The final stage is to try to reallocate some of the parts of blocks that have been split in an attempt to reduce the cost of late jobs at the expense of an extra respin. We work with the blocks in the same order that they were assigned to lines. Suppose we are considering reallocating parts of block B_i. Let $\hat{\pi}_{ij}$ denote the proportion of B_i that would have to be removed from line L_j in order that all jobs on that line were finished by the end of the week. Then

$$\hat{\pi}_{ij} = \min \left\{ \max \left\{ 0, \frac{r_j(-a_j - 1)}{s_i r_j + l_i} \right\}, \pi_{ij} \right\}.$$

The minus one is due to the fact that a_j has already been updated to take care of a potential respin occurring after the last block. The cost $\hat{c}_j$ of removing proportion $\hat{\pi}_{ij}$ of the block can easily be calculated.

We try to reallocate parts of block B_i to the line L_j belonging to F_i with the greatest value of

$$\frac{r_j a_j}{s_i r_j + l_i}.$$

Denote this line by L_k. Now order the lines with $\hat{\pi}_{ij} > 0$ in decreasing order of $\hat{c}_j / \hat{\pi}_{ij}$ and from each in turn remove as large a proportion of B_i up to a maximum of $\hat{\pi}_{ij}$ and allocate it to line L_k until the actual capacity of line L_k is used up. It is straightforward to compare the improvement in the cost because fewer jobs are late with the cost of the extra respin and decide whether the reallocation is worthwhile. Should such a reallocation prove to be worthwhile we repeat the process and attempt to reallocate again until it is no longer worthwhile.

Following the above procedure, each leaf node of the DFS tree corresponds to an assignment of all the blocks to the lines. For each leaf node of the DFS tree, the cost of the assignment is estimated by summing the estimated cost due to late jobs on each line and the costs due to respins. At the end of the procedure the assignment with the least cost is selected as the one used.

If no assignment satisfying the useful capacities is found during the DFS then the process is repeated with u_j incremented by $1/(0.066\alpha r_j)$ for each j repeatedly until a feasible assignment is found.

4.2 Sequencing within a block

In this section we begin by describing an algorithm that will sequence the jobs within a block assuming that the block is all to be processed on a single line. Later we will discuss how this algorithm can be used to split long blocks within two lines.

We assume that we have jobs $J_1, \ldots, J_n$ to be sequenced. Each job J_i has a processing time p_i, a weight w_i and a due time d_i. Notice that the weights are proportional to the processing times with the constant of proportionality depending on the speed of the line and the kilotex. Now let $c_{i,j}$ denote the time to change the set-up of the line between the configuration required for job J_i and that required for job J_j. In order to simplify the algorithm we first compute a set of jobs for which the due dates are guaranteed to be met using any schedule.

Suppose we start processing the block at time T_0. The time to process the complete sequence of jobs is at most

$$M = \sum_{i=1}^{n} p_i + \sum_{i=1}^{n} \max\{c_{i,j} : 1 \leq j \leq n\}.$$

Therefore those jobs with due time d_i satisfying $d_i + T_0 \geq M$ will not be late under any schedule and so they are given a new due time of ∞ reflecting the fact that the due date will not concern us.

The problem is now to find a permutation $\pi : [1, n] \to [1, n]$ (which we regard as a sequence of the jobs $J_1, \ldots, J_n$) that minimises

$$\sum_{i=1}^{n} p_i + \sum_{i=1}^{n-1} c_{\pi(i),\pi(i+1)} + \alpha \sum_{i=1}^{n} w_i I(\pi, i),$$

where $I(\pi, i)$ is an indicator variable taking the value one if job J_i is late under the schedule given by π and the value zero otherwise.

Our problem is very similar to the well-studied travelling salesman problem (TSP) [2, 4, 5, 6, 9, 10]. In the TSP we are given a complete graph on vertices $\{v_1, \ldots, v_n\}$ where the edge (v_i, v_j) has weight $a_{i,j}$ and we seek a permutation π of the vertices which minimises

$$\sum_{i=1}^{n} a_{\pi(v_i),\pi(v_{i+1})},$$

where by $\pi(v_{n+1})$ we mean $\pi(v_1)$. In the case where all due dates are equal to infinity then by adding a dummy job J_0 with processing time zero, due date ∞ and set up costs $c_{0,i} = c_{i,0} = 0$ for all i, we transform the problem to an instance of the TSP. Unfortunately the TSP is an NP-hard problem and so it is extremely unlikely that there is a fast, efficient (*i.e.* polynomial time) algorithm to solve it. However a huge range of heuristics has been studied for the TSP and the similarity with the sequencing problem suggests that these heuristics might be suitable for solving the sequencing problem.

Although there is no analytical method to formally determine the best heuristic for the TSP, the best methods appears to be branch and cut algorithms using integer programming formulations [2, 9]. It does not appear to be straightforward to make these methods work when we have extra constraints such as those given by the due dates. Consequently we have opted for an approach using construction heuristics followed by local search. A construction heuristic for a scheduling problem builds up a schedule step by step, adding a new job into the schedule at each stage. There is a rule for selecting the next job to be inserted and a second rule for deciding where to insert it into the current partial schedule. One construction heuristic that appears to have relatively good performance is farthest insertion [10] which we describe now.

Within each block, we first partition the jobs into two sets, A_1 and A_2, where A_1 contains those jobs J_i for which d_i is at or before the end of the current week and A_2 contains the others. Notice that because we are only sequencing a week's worth of jobs and we assume that all the due dates correspond to the end of some week, the jobs in A_1 are precisely those jobs for which the choice of how the block is scheduled will determine whether they are late or not. Furthermore A_1 will be empty if the choice of schedule has no effect on which jobs are late.

The next stage is to sequence the jobs in A_1. We assume that

$$A_1 = \{J_1, J_2, \dots J_m\}.$$

We begin by finding the pair of jobs J_r and J_s which achieve the maximum in

$$\max_{i \neq j} \min\{c_{i,j}, c_{j,i}\}$$

and forming an initial sequence from them by taking them in the order achieving the minimum.

Throughout the process we build up a partial schedule by inserting one job at a time. Suppose, without loss of generality, that the schedule assembled so far is $J_1, J_2, J_3, \dots, J_k$. For each job J_l where $k+1 \leq l \leq m$ we compute the minimum extra delay m_l resulting from inserting J_l into one of the possible $k+1$ positions in the schedule. The extra delay incurred from inserting J_l between J_r and J_s is

$$c_{r,l} + c_{l,s} - c_{r,s}.$$

Since the costs satisfy the triangle inequality, this value is always positive. We then find the unplaced job with the maximum value of m_l and add it to the schedule in the place where this value is achieved, that is the place where the minimum extra delay is incurred for that job.

Notice that at each stage in this process, we ignore the extra cost incurred from jobs being late. This is reasonable providing the jobs all have similar processing times, for then we have little control over how many jobs are late and can make the reasonable assumption that the schedule requiring the smallest total time is a good one.

The quality of the solution can be improved by applying a procedure such as 3-OPT. This is an iterative local improvement heuristic. To describe one stage of the iteration, suppose without loss of generality that in the current schedule the jobs are ordered $J_1, J_2, \ldots, J_n$. The schedule J is broken into four pieces and the schedule $\hat{J}$ produced by interchanging the intermediate pieces is compared with the original schedule. If $\hat{J}$ has cost less than that of J then we replace J by $\hat{J}$ and start another iteration, otherwise we carry on choosing places to break the chain. More precisely we enumerate all possible choices of integers j, k and l such that $1 \leq j < k < l \leq n+1$ and find the cost of the schedule $\hat{J}$ given by taking the jobs in the order

$$J_1, \ldots, J_{j-1}, J_k, J_{k+1}, \ldots, J_{l-1}, J_j, J_{j+1}, \ldots, J_{k-1}, J_l, \ldots, J_n.$$

This procedure is repeated until either a local optimum is reached or a fixed number of iterations have been carried out. Using an implementation [4], which we do not describe here, the procedure can be implemented so that each iteration takes time $O(n^2)$.

Once the jobs in A_1 have been sequenced, we take any jobs from A_1 that will be late in the schedule and then add them to A_2. We now sequence A_2 in the same way as for A_1 producing a sequence to follow the one that we have already constructed.

For the set of jobs discussed in the Section 3, the set-up times are determined by the matrix (29). If we take the (i,j) entry to be the time taken in minutes to change colour from that required for job J_i to that required for job J_j where $\{J_1, \ldots, J_7\}$ is a set of jobs forming a block for which due dates are sufficiently large to be dismissed, the sequence produced by the farthest insertion heuristic is $J_6, J_3, J_2, J_1, J_5, J_7, J_4$ with a total set-up time of 21.56 minutes. Careful checking shows that this is optimal and that the 3-OPT procedure would not even be needed.

Generally the farthest insertion heuristic will not produce the optimal solution. Empirical evidence [4] suggests that for problems without due dates, it produces a solution with cost about 10-15% above the optimal value. Again empirical evidence suggests that 3-OPT can achieve a significant improvement on this and will tend to produce a solution with cost about 3-4% above the optimum.

Split blocks. Blocks which need to be split over two or more lines have their jobs sequenced last. We first obtain upper bounds b_j on the amount of time that can be used for processing the jobs from B_i allocated to line L_j. Suppose that the DFS has allocated a proportion π_{ij} of the jobs from B_i to line L_j. An estimate of the total time needed to process the proportion of B_i allocated to L_j is

$$\pi_{ij}(s_i + l_i/r_j).$$

If this estimate does not exceed the remaining actual capacity of L_j and there are no other portions of split blocks on line L_j still to be sequenced then we set b_j to be the remaining actual capacity of L_j (after all the blocks other than B_i have been sequenced). Otherwise we set

$$b_j = \pi_{ij}(s_i + l_i/r_j).$$

When the farthest insertion heuristic is run it first selects the initial jobs to be assigned to each machine by choosing those jobs which maximise $\min_j c_{i,j}$. One of these jobs is allocated to each line. The procedure then carries on exactly as before except that the heuristic will never insert a job into a line which would cause the total processing time to exceed b_j, unless it becomes impossible to prevent this.

When a block has been split between lines, the 3-OPT heuristic can carry out two types of move. It can interchange segments of jobs within a portion of a block or it can interchange two segments of jobs between two portions of a block. Without loss of generality suppose a block is split between lines $L_1, \ldots, L_k$. Suppose further that there are m_i jobs assigned to L_i. The 3-OPT routine first makes a choice of line, L_j say, and then chooses two integers p and q with $1 \leq p < q \leq m_j + 1$. It may then do one of two things. Firstly it may choose a third integer r such that $q < r \leq m_j + 1$ and carry out a normal 3-OPT move as before or it may choose another line m'_j and an integer r such that $1 \leq r \leq m'_j - q + p + 1$ and consider the benefit of interchanging the segments of jobs $J_{j,p}, \ldots, J_{j,q-1}$ with $J_{j',r}, \ldots, J_{j',r+q-1-p}$. As before 3-OPT can be run for a fixed period of time or until a local optimum is reached.

4.3 Sequencing over more than 1 week

In this section we describe the additional complications that result from having a rolling schedule. The procedure we describe will produce a schedule looking at a two-week window. We assume that for each week we are given a collection of new orders, some of which must be completed by the end of the week. Much of the procedure is exactly as described above and we work with a week's worth of jobs at a time. However we allow ourselves the option of altering the schedule for the previous week as we construct the next week.

We divide the new jobs into blocks in exactly the same way as before and discard any blocks with no jobs to be assigned by the end of the week as before. Actual and useful capacities are defined in a similar way to before except that they have to take account of the time that the lines finish processing the jobs from the previous week's schedule. This is done by adding the remaining actual capacity of the previous week's jobs to the initial values of actual and useful capacity.

The DFS proceeds in a similar fashion to before except in two places. When it attempts to add a block B_i to a line it may either assign it as a new block in its own right or if there is a compatible block $\hat{B}$ scheduled in the previous week then it will consider the possibility of combining the two blocks by inserting the new block immediately after $\hat{B}$, avoiding the cost of an extra respin. (Two blocks are said to be compatible if their jobs may be considered as a single set, still without incurring any respin.) When doing this, the extra cost due to jobs from the previous week being late will be compared with the cost of an extra respin. Secondly when the DFS attempts to order the blocks on a line it will also try to reorder the blocks from the previous week if a pair of blocks, one from each week, are compatible. This may produce an improved schedule if the ordering can be arranged so that the last block from the first week is compatible with the first block of the second week, avoiding the need for a respin.

The farthest insertion and 3-OPT heuristics work in the same way as for when just one week is scheduled except in the case when there is a set of compatible blocks with one or more in each week of the schedule. In this case both the heuristics are modified to work in a similar way to how they work when a block has to be split between lines. The farthest insertion heuristic is allowed to add jobs to blocks in the previous week with the proviso that no job in any block in the previous week may become late as a result of doing this. The 3-OPT heuristic will attempt to move sequences of jobs between blocks, again in a similar way to when a block has been split.

5 Conclusions

In our model of the problem, some simplications have been made and it is important to understand the implications of these. The main simplification is the way we have treated due dates. In a more detailed model, the two different types of due dates that we have described could be dealt with more accurately or a function giving the penalty for lateness for each job could be specified.

Our solution method has two stages. In the first part the jobs are partitioned into blocks and the blocks assigned to lines. This task seems relatively straightforward for the two sets of real data we have seen but we do not know whether this is generally the case. In the second stage sequencing is carried out within the blocks. This uses methods developed for the Travelling Salesman Problem, for which there is a vast literature. The problem considered here is complicated by due dates, meaning that some of the methods for the Travelling Salesman Problem cannot be used or are much less efficient. The method we have chosen to use is generally regarded as being one of the better heuristics for the Travelling Salesman Problem but not the best. However it can cope with the problem of due dates. It is possible to refine the method to achieve significant speed-up when sequencing a set of jobs for which any sequence meets the due dates, but it is unclear whether this refinement can be achieved when due dates are important. An interesting question is to determine how much time is spent in each part of the algorithm and which parts are the most efficient.

To evaluate the effectiveness of the methods we have discussed, it will be necessary to schedule a number of weeks of jobs and compare with the schedules currently produced by Acordis. This will be particularly useful to identify desirable features that have not been incorporated within the model. A reasonable approach to combining the computer and human schedulers would be to use the computer for assigning jobs within a block. This is a simple problem to describe but a hard problem to solve. In practice there seems less scope for finding good solutions to the problem of assigning blocks to lines. Because of the nature of the constraints in this part of the problem, to ensure that the computer does not miss a solution which is fairly obvious to a human requires the construction of a cumbersome and only moderately efficient program.

References

[1] *Basic parameter values for the HDTV standard for the studio and for the international program exchange*, ITU-R Recommendation BT.709, ITU, Geneva (1991).

[2] D. L. Applegate, R. Bixby, V. Chvátal and W. Cook, *Finding cuts in the TSP*, Technical report, DIMACS (1995).

[3] S. D. Howison and R. J. Lawrence, *Fluorescent transfer of light in dyed materials*, SIAM J. Appl. Math., **53**(2) 447–458 (1993).

[4] D. S. Johnson and L. A. McGeoch, *The traveling salesman problem: a case study in local optimization*, in E. H. L. Aarts and J. K. Lenstra (editors) *Local Search in Combinatorial Optimization*, pp. 215–310, John Wiley, 1997.

[5] S. Lin, *Computer solutions of the traveling salesman problem*, Bell Systems Technical Journal, **44** 2245–2269 (1965).

[6] S. Lin and B. W. Kernaghan, *An effective heuristic algorithm for the traveling-salesman problem*, Operations Research, **21** 498–516 (1973).

[7] D. L. MacAdam, *Visual sensitivities to color differences in daylight*, J. Optical Soc. America, **32** 247–274 (1942).

[8] L. McLaren, *The Color Science of Dyes and Pigments*, Adam Hilger, Bristol, 1983.

[9] M. Padberg and G. Rinaldi, *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*, SIAM Review, **33** 60–100 (1991).

[10] D. J. Rosenkrantz, R. E. Stearns and P. M. Lewis, *An analysis of several heuristics for the traveling salesman problem*, SIAM J. Comput., **6** 563–581 (1977).

[11] G. Wyszecki and W. S. Stiles, *Color Science, Concepts and Methods, Quantitative Data and Formulae* (Second edition), John Wiley, 1982.