MATH2730 - Operations Research I

Group Project

Due: 4pm Friday 4th May 2018

Contribution to Final Grade: 20%

The decision problem at the core of this case is the scheduling of weekly production of five parts on five machines, where each part can be produced only on a subset of the machines. Each time the production on a machine changes from one part to another, a significant amount of machine time is spent on set-up. Regular time production capacity is insufficient to meet the demand and overtime production is generally necessary. This problem, and the subsequent extension, is based on a case “Production Scheduling at Falcon Die Casting” of B. Madhu Rao (Stetson U.) and Jeroen Beli ¨en (KU Leuven). See the corresponding slides for the full details and data

1. [Marks: 4] Write down an integer linear program (ILP) that decides what manufacturing policy should the company pursue in order to minimize the sum of overtimes scheduled on the five machines. Make your model as general as possible. Make sure you clearly define all your variables, and briefly explain the purpose of each constraint, and/or component of the objective function.
Implement your linear programming model in XPRESS and use XPRESS to solve the linear program for the values giving in the description above. Make use of XPRESS Mosel functions to produce a report of the results.

There are 5 parts and 5 machines, which are referred to with i and j respectively when data is stored for specific parts and/or machines in 1 or 2 dimensional arrays.

Variables:

u is the maximum number of hours that the machines can spend working each week for the part of the week which costs the "regular" amount, which in this case is every hour of a weekday. So

u = 120

Since there are 120 hours of weekdays in a week.

v is the maximum number of hours that the machines can spend in "overtime" each week, which is every hour of the weekend. So

v = 48

Since there are 48 hours of weekend days in a week.

$\mathbf{b_i}$ is the demand for each part for a predetermined week, essentially how much needs to be produced. The demand for the first week is used, which is:

Demand	Part 1	Part 2	Part 3	Part 4	Part 5
Week 1	3500	3000	4000	4000	2800

For each respective part.

$\mathbf{s_{ij}}$ is the number of hours it takes to setup a specific machine for the production of a specific part. The setup times are:

Parts	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Part 1	8	10			
Part 2		8	10	8	
Part 3				12	
Part 4	8				8
Part 5			24		20

(Empty cells represent machines that can't make the specific part, and so shouldn't or can't be setup to produce them.)

$\mathbf{r_{ij}}$ is the number of parts, of a specific type, a specific machine can produce in an hour, the production rate for each part/machine combination. The production rates are:

Parts	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Part 1	40	35	0	0	0
Part 2	0	25	30	35	0
Part 3	0	0	0	50	0
Part 4	60	0	0	0	60
Part 5	0	0	45	0	50

(Cells with 0 have a production rate of 0, meaning a machine can't make that part.)

$\mathbf{y_i}$ is the yield for each part, the percentage of the parts that are produced that meet quality specifications. The yields for each part are:

	Part 1	Part 2	Part 3	Part 4	Part 5
Yield	0.60	0.55	0.75	0.65	0.60

The yield would then be applied to $\mathbf{r_{ij}}$, the production rate, to give the actual production rate, of parts that meet the quality specifications, shown in the table below:

$\mathbf{r_{ij}}$	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Part 1	24	21	0	0	0
Part 2	0	13.75	16.5	19.25	0
Part 3	0	0	0	37.5	0
Part 4	39	0	0	0	39
Part 5	0	0	27	0	30

$\mathbf{M_{ij}}$ is the maximum number of a specific part a specific machine could produce in a week, to be used for the big-M constraints that set $\mathbf{X_{ij}}$. The formula for this is,
$\mathbf{M_{ij}} = \mathbf{r_{ij}} * (\mathbf{u}+\mathbf{v}-\mathbf{s_{ij}})$ which produces these values:

	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Part 1	3840	3318	0	0	0
Part 2	0	2200	2607	3080	0
Part 3	0	0	0	5850	0
Part 4	6240	0	0	0	6240
Part 5	0	0	3888	0	4440

(Cells with have a production rate of 0, meaning a machine can't make that part.)

Decision Variables:

Integer Decision Variables:

$\mathbf{P_{ij}}$ is the number of parts produced by each machine of each part. This can be used to determine how much time the machine spent in operation, and so also how much time it spent in overtime, allowing us to sum all of the overtime in the objective function and minimise that. $\mathbf{P_{ij}}$ must be an integer because a fraction of a part can't be made and wouldn't have value.

Binary Decision Variables:

$\mathbf{X_{ij}}$ represents the decision of whether a given machine will be used to produce a specific part. It allows the setup time for producing each type of part on a given machine to be included in that machine's time constraint. It is 1 when a machine has produced any number of that specific part and 0 otherwise, and is set by the big-M constraints. Without $\mathbf{X_{ij}}$ the overtime would be significantly underestimated and the production schedule almost certainly incorrect.

Real-Valued/Regular Decision Variables:

$\mathbf{O_j}$ is the number of hours a machine will spend in overtime to produce the required amount. The sum of these values is the objective function that needs to be minimised for Question 1.

Objective Function:

OvertimeSum = min $\sum_{j=1}^{5}$ **O_j**

This objective function minimises the sum of overtimes scheduled on each of the (5) machines.

Constraints:

$For\ all\ j \in \{1...5\}$

$TimeCS_j$:= $(\sum_{i=1}^{5} \frac{Pij}{rij} + X * s) \leq u + O$

When r=0 because the machine can’t make that part, that iteration won’t be included in the sum for the xpress program to avoid dividing by 0, though P and X will also equal 0 so when fully written out those instances would be 0 anyway.

The time constraint limits the amount being produced by a machine per week to what is possible given the number of hours in that week. Without this constraint, overtime would be misreported as 0 hours for a given production schedule despite overtime potentially happening.

The time the machine takes is the sum of all time taken to produce all parts including the time it takes to setup the machine for the production of each part. This is constrained by the maximum regular time and the overtime, where the overtime is separately constrained by the maximum overtime, which gives a maximum time of 168 hours in a week.

$For\ all\ i \in \{1...5\}$

$DemandCS_i$:= $(\sum_{j=1}^{5} Pij) \geq bi$

The demand constraint constrains the number of parts produced to at least the demand for the week. For this case, it’s possible to meet the demand for the first week, so another variable to deal with demand that hasn’t been met, a shortfall in production, hasn’t been included. That is covered in question 4. Without this constraint, production and overtime would fall to 0 for question 1.

There is a demand constraint for each part, where the sum of all machine’s production of that part must be at least the demand for that part.

$$For\ all\ i \in \{1...5\}\ \&\ all\ j \in \{1...5\}$$

MCS_{ij} := $Pij \leq (Mij * Xij)$

The big-M constraint ensures that Xij is 1 when production is greater than 0, and 0 otherwise. To ensure that the time taken to set up a machine is included in the time constraint whenever that machine is actually producing the given part.Since M is the highest P can be, this will always work for any possible value of P. Without this constraint the setup time wouldn't be included in the time constraint.

There is a separate constraint for every P value since there are different production rates and setup times for each combination of part and machine.

$$For\ all\ i \in \{1...5\}\ \&\ all\ j \in \{1...5\}$$

OCS_j := $Oj \leq v$

The O constraint ensures that overtime doesn't exceed the maximum value it can be. Admittedly this isn't relevant to this question because the demand would take less time to produce than the maximum time available in a week, and the objective is to minimise O while filling the demand, but in cases where the demand was greater, this would limit the value to what is possible, 48 hours for the overtime period of the weekend.

See Results/Solution Report in Appendix.

2. [Marks: 3] Whenever overtime production needs to be scheduled, the traditional practice has been to schedule it on machines that are most efficient for the part being produced. This has often led to uneven overtime assignment in the sense that long hours of overtimes are scheduled on one or two machines while other machines remained idle. While this resulted in lower total overtime paid to production personnel, it often resulted in higher overall costs because of the overtime costs of the required support personnel such as administrative assistants, electricians, material handlers and quality control technicians. Their presence is necessary as long as production is in process, irrespective of the number of machines operating. Tom wondered if it would be more economical to schedule production on more machines over the weekend and minimize the total duration for which overtime production takes place.
Modify your model to minimize the total duration of overtime production (i.e., maximum of the overtimes on all machines) rather than the sum of overtimes on all machines. Compare the optimal production schedule obtained with the new objective with that for Question 1 and then discuss the nature of changes in the optimal solution.

Include all previous variables and constraints.

Real-Valued/Regular Decision Variables:

LargestHours is the decision variable for what the largest or maximum of all the Oj values is. It will be minimised in the objective function and set by constraining all Oj values to being at most this variable.

Objective Function:

Min LargestHours_j

This objective function minimises the largest of all the overtimes for each machine, the Oj values. Another way of putting this is minimising the time the entire facility spends in operation on the weekend because at least one machine is still running.

Constraints:

$$For\ all\ j \in \{1...5\}$$

OMCS_j := $Oj \leq LargestHours$

This constraint ensures that LargestHours is equal to the largest of the Oj values, as LargestHours will be minimised while the Oj values will still try to fulfill the other time constraints.

Compare production schedules, discuss nature of changes in optimal solution.
(Note, see appendix for production schedules, and optimal solutions in general for Q1 and Q2).
For this production schedule, the sum of all overtime hours is 136 hours, compared to 119.77 for Q1, potentially increasing the cost of running the business if the main costs are per overtime hour. However the maximum overtime hours for a machine is 27.2 compared to almost 48 in Q1, which allows for a much more even spread of wear and use on the machines, and allowing all machines time to be repaired or preventatively maintained. This could also bring down costs if the main costs from overtime are from having the facility open and support staff present at all. As is shown in Q3, a production schedule somewhere between the two is more likely to be relevant and cost effective.
To choose the Q2 objective over the Q1 objective you would need a formula like this:
Difference in Max Overtimes * Facility Running Costs
Would have to be greater than:
Difference in Sum of Overtime Hours * Cost per Hour of Overtime
(48-27.2)*RunCost > (136-119.77)*OvertimeCost
The nature of the changes to the optimal solution for the production schedule are most clearly visible with part 2 machine 2 and part 4 machine 5, where previously in Q1 they were 0. In general the optimal solution has spread out the production across more machines, despite the additional setup times, so that more machines are working at once to achieve a lower maximum time. This is clearly demonstrated with all the overtime's being almost exactly the same, as intended by the objective function.

3. [Marks: 2] Modify your model to minimize the total cost of overtime for production and support personnel , assuming that the cost of scheduling overtime on each machine is $30 per hour and the cost of support personnel during overtime is $40 per hour.

The model in question 3 builds on top of the model we had used in question 2. We re-use *LargestHours* here (which is the maximum amount of overtime hours we have spent (ie max(overtime hours) where overtime hours = {oi, o2, o3, o4, o5}).

We also introduce 2 new scalar variables:

- C_1 : cost of overtime production per hour per machine ($30)
- C_2 : cost of overtime support (via personnel) per hour ($40)

Note: overtime on each machine happens concurrently so the cost of overtime support is concerned with the maximum amount of time overtime might be scheduled for. For example machines 1, 2, 3 and 4 are scheduled for 4 hours of overtime and machine 5 is scheduled for 10 hours of overtime so the overtime support personnel need to be rostered on for 10 hours.

Objective Function: $\textbf{Min}\ c_1 \cdot (\sum_{i=1}^{5} O_i) + c_2 \cdot LargestHours$

4. [Marks: 3] FDC is strongly committed to being responsive to customer's needs and attempts to satisfy all demand by using overtime when necessary. On some occasions, weekly production requirements exceeded FDC's capacity, even with maximum overtime. Under these conditions, FDC believes that it is critical that these shortfalls are recognized early in the process and communicated to the customer as soon as possible with a clear indication of the likely delays in delivery. Modify your model to determine the optimal production schedule and also generate information on potential shortfalls. Use the delivery schedule for week 1 in Table 5 in this case and assume that a penalty cost of $3 per week is imposed for each unit not delivered on time. For this model, assume that the cost of carrying inventory is $2 per unit per week.

Include all previous variables and constraints, except the demand variable and demand constraint which will be replaced.

Variables:

C3 := 3 is the cost ($3) incurred for each part not produced that was demanded.

$\mathbf{b_i}$ is the demand for each part for a predetermined week, essentially how much needs to be produced. The demand for the first week from Table 5 is used, which is:

Demand	Part 1	Part 2	Part 3	Part 4	Part 5
Week 1	4500	4000	5000	5000	3800

For each respective part.

Decision Variables:

Integer Decision Variable:

$\mathbf{U_i}$ is the unsatisfied demand for each part, the number of parts that were demanded but couldn't be produced. The sum of all unfulfilled demands is minimised in the objective function.

Objective Function:

UnsatisfiedDemandCost := minimise

$$(c3 * \sum_{i=1}^{5} U_i) + (c1 * \sum_{j=1}^{5} O_j) + (c2 * LargestHours)$$

This objective function minimises the cost of production from overtime and from unsatisfied demand, adding to the objective function from question 3, given a demand that cannot be completely met in the given week.

Constraints:

$For\ all\ i \in \{1...5\}$

DemandCS$_i$:= $Ui + \sum_{j=1}^{5} Pij \ \geq bi$

To handle the new demand values/variable, a new demand constraint is required to account for what would have previously been an infeasible solution.
The demand constraint constrains the number of parts produced to at least the demand for the week. For this case, it’s not possible to meet the demand for the first week from Table 5, so another variable to deal with demand that hasn’t been met, a shortfall in production, has been included, for question 4. Without this constraint, production and overtime would fall to 0 for question 1.
There is a demand constraint for each part, where the sum of all machine’s production of that part must be at least the demand for that part, the unsatisfied demand U is added to production, and will be minimised, such that it will be equal to the unsatisfied demand, making up the shortfall P leaves.

Hand in a report that presents and discusses your models and that includes as appendices a print-out of the XPRESS implementation of each model as well as a print-out of the solution report produced by your XPRESS implementation. [Report Marks: 8]

Appendices

Question 1

XPRESS Implementation

```
!@encoding CP1252
model FalconDieCasting
uses "mmxprs"; !gain access to the Xpress-Optimizer solver

declarations
	NUMMACHINES = 5
	NUMPARTS = 5
	NUMWEEKS = 2
	MACHINERANGE = 1..NUMMACHINES		!j
	PARTRANGE = 1..NUMPARTS		!i
	u: real !maximum regular time per week
	v: real !maximum overtime per week

	b: array(PARTRANGE) of real !demand for each part
	s: array(PARTRANGE, MACHINERANGE) of real !setup time for producing each part on each machine
	y: array(PARTRANGE) of real !percentage yield of useable parts for each part
	r: array(PARTRANGE, MACHINERANGE) of real !production rate of each part on each machine, needs to be initialised
and then * by yield

	M: array(PARTRANGE, MACHINERANGE) of real !big M, max amount P could be

	!DECISION VARIABLES
	P: array(PARTRANGE,MACHINERANGE) of mpvar !amount produced of each part on each machine
	X: array(PARTRANGE, MACHINERANGE) of mpvar !binary, machine used to make part
	O: array(MACHINERANGE) of mpvar	!Overtime for each machine

	!OBJECTIVE FUNCTION
	OvertimeSumOF: linctr !Question 1: Sum of all overtime hours

	!CONSTRAINTS
	TimeCS: array(MACHINERANGE) of linctr
	DemandCS: array(PARTRANGE) of linctr
	MCS: array(PARTRANGE, MACHINERANGE) of linctr
	OCS: array(MACHINERANGE) of linctr

end-declarations
	!initialise all constants/variables
	u := 120.0 !maximum regular time per week
	v := 48.0 !maximum overtime per week

	b :: [	3500,	3000,	4000,	4000,	2800] !demand for each part
	s :: [	8,		10,		999,	999,	999,!setup time for producing each part on each
machine
			999,	8,		10,		8,		999,
			999,	999,	999,	12,		999,
			8,		999,	999,	999,	8,
			999,	999,	24,		999,	20]
```

```
	r :: [	40.0,	35.0,	0.0,	0.0,	0.0, !production rate initial values
			0.0,	25.0,	30.0,	35.0,	0.0,
			0.0,	0.0,	0.0,	50.0,	0.0,
			60.0,	0.0,	0.0,	0.0,	60.0,
			0.0,	0.0,	45.0,	0.0,	50.0]

	y :: [0.6,	0.55, 0.75, 0.65, 0.6]

	!initial calculations, set to binary or integer
	forall(j1 in 1..NUMMACHINES) do
		forall(i1 in 1..NUMPARTS) do
			P(i1,j1) is_integer	!make P, amount produced an integer vlaue
			X(i1,j1) is_binary	!make X, whether or not started
			r(i1,j1) := r(i1,j1)*y(i1)		!calculate r, taking into account yield of each machines
			M(i1,j1) := r(i1,j1)*(u + v - s(i1,j1))
		end-do
	end-do

	!OBJECTIVE FUNCTION QUESTION 1
	OvertimeSumOF := sum(j2 in 1..NUMMACHINES) O(j2)

	!TIME CONSTRAINTS
	forall(j3 in 1..NUMMACHINES) do
		!timetaken(j3) := sum(i3 in 1..NUMPARTS) (P(i3,j3)/r(i3,j3) + X(i3,j3)*s(i3,j3))
		TimeCS(j3) := sum(i3 in PARTRANGE | r(i3,j3)>0) (P(i3,j3)/r(i3,j3) + X(i3,j3)*s(i3,j3)) <=  u + O(j3)
	end-do

	!DEMAND CONSTRAINTS
	forall(i4 in 1..NUMPARTS) do
		DemandCS(i4) :=	(sum(j4 in 1..NUMMACHINES) P(i4,j4)) >= b(i4) !demand CS for 2nd week
	end-do

	!BIG M CONSTRAINTS
	forall(i5 in PARTRANGE) do
		forall(j5 in MACHINERANGE) do
			MCS(i5,j5) := P(i5,j5) <= M(i5,j5)*X(i5,j5)
		end-do
	end-do

	!MAX OVERTIME CONSTRAINT
	forall(j6 in MACHINERANGE) do
		OCS(j6) := O(j6) <= v
	end-do

minimise(OvertimeSumOF) !Question1
writeln("Question 1 Objective is Overtime Sum =", getact(OvertimeSumOF))
write("Overtime hours: ")
forall (i in 1..NUMMACHINES) do
	write(getsol(O(i)))
	write(", ")
end-do

end-model
```

Question 1: Solution Report:

Objective: Minimise Sum of all Overtimes = 119.773

Decision Variables:

	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Overtime (hrs)	47.9808	0.190476	23.6364	47.9654	2.55843e-12

Production Schedule	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Part 1	1186	2314	0	0	0
Part 2	0	0	2205	795	0
Part 3	0	0	0	4000	0
Part 4	4000	0	0	0	0
Part 5	0	0	0	0	2800

Question 2

XPRESS Implementation:

```
!@encoding CP1252
model FalconDieCasting
uses "mmxprs"; !gain access to the Xpress-Optimizer solver

declarations
	NUMMACHINES = 5
	NUMPARTS = 5
	NUMWEEKS = 2
	MACHINERANGE = 1..NUMMACHINES		!j
	PARTRANGE = 1..NUMPARTS		!i
	u: real !maximum regular time per week
	v: real !maximum overtime per week

	b: array(PARTRANGE) of real !demand for each part
	s: array(PARTRANGE, MACHINERANGE) of real !setup time for producing each part on each machine
	y: array(PARTRANGE) of real !percentage yield of useable parts for each part
	r: array(PARTRANGE, MACHINERANGE) of real !production rate of each part on each machine, needs to be initialised
and then * by yield

	M: array(PARTRANGE, MACHINERANGE) of real !big M, max amount P could be

	!DECISION VARIABLES
	P: array(PARTRANGE,MACHINERANGE) of mpvar !amount produced of each part on each machine
	X: array(PARTRANGE, MACHINERANGE) of mpvar !binary, machine used to make part
	O: array(MACHINERANGE) of mpvar	!Overtime for each machine

	!OBJECTIVE FUNCTION
	LargestHours:mpvar !Question 2: Largest number of hours for one machine


	!CONSTRAINTS
	TimeCS: array(MACHINERANGE) of linctr
	DemandCS: array(PARTRANGE) of linctr
	MCS: array(PARTRANGE, MACHINERANGE) of linctr
	OCS: array(MACHINERANGE) of linctr
	OMCS: array(MACHINERANGE) of linctr

end-declarations
	!initialise all constants/variables
	u := 120.0 !maximum regular time per week
	v := 48.0 !maximum overtime per week

	b :: [	3500,	3000,	4000,	4000,	2800] !demand for each part

	s :: [	8,		10,		999,	999,	999,!setup time for producing each part on each
machine
			999,	8,		10,		8,		999,
			999,	999,	999,	12,		999,
			8,		999,	999,	999,	8,
			999,	999,	24,		999,	20]

	r :: [	40.0,	35.0,	0.0,	0.0,	0.0, !production rate initial values
			0.0,	25.0,	30.0,	35.0,	0.0,
			0.0,	0.0,	0.0,	50.0,	0.0,
```

```
				60.0,	0.0,	0.0,	0.0,	60.0,
				0.0,	0.0,	45.0,	0.0,	50.0]

		y :: [0.6,   0.55, 0.75, 0.65, 0.6]

		!initial calculations, set to binary or integer
		forall(j1 in 1..NUMMACHINES) do
			forall(i1 in 1..NUMPARTS) do
				P(i1,j1) is_integer	!make P, amount produced an integer vlaue
				X(i1,j1) is_binary	!make X, whether or not started
				r(i1,j1) := r(i1,j1)*y(i1)		!calculate r, taking into account yield of each machines
				M(i1,j1) := r(i1,j1)*(u + v - s(i1,j1))
			end-do
		end-do

		!TIME CONSTRAINTS
		forall(j3 in 1..NUMMACHINES) do
			!timetaken(j3) := sum(i3 in 1..NUMPARTS) (P(i3,j3)/r(i3,j3) + X(i3,j3)*s(i3,j3))
			TimeCS(j3) := sum(i3 in PARTRANGE | r(i3,j3)>0) (P(i3,j3)/r(i3,j3) + X(i3,j3)*s(i3,j3)) <=  u + O(j3)
			!writeln("Time Constraint(",j3, ") is:", TimeCS(j3) )
		end-do

		!DEMAND CONSTRAINTS
		forall(i4 in 1..NUMPARTS) do
			!DemandCS(i4) :=   sum(j4 in 1..NUMMACHINES) P(i4,j4) >= b(1, i4) !demand CS for 1st week
			DemandCS(i4) :=    (sum(j4 in 1..NUMMACHINES) P(i4,j4)) >= b(i4) !demand CS for 2nd week
		end-do

		!BIG M CONSTRAINTS
		forall(i5 in PARTRANGE) do
			forall(j5 in MACHINERANGE) do
				MCS(i5,j5) := P(i5,j5) <= M(i5,j5)*X(i5,j5)
			end-do
		end-do

		!MAX OVERTIME CONSTRAINT
		forall(j6 in MACHINERANGE) do
			OCS(j6) := O(j6) <= v
		end-do

		!MAX MACHINE OVERTIME CONSTRAINT Objective for Question 2
		forall(j7 in MACHINERANGE) do
			OMCS(j7) := O(j7) <= LargestHours
		end-do

	minimise(LargestHours)		!Question 2
	writeln("Question 2 Objective is Overtime Maximum = ", getsol(LargestHours))
	write("Overtime Hours: ")
	forall(j8 in MACHINERANGE) do
		write(getsol(O(j8)))
		write(", ")
	end-do
	writeln
	forall (i in 1..NUMMACHINES) do
		forall (j in 1..NUMPARTS) do
			write(getsol(P(i,j)))
			write(", ")
```

```
        end-do
        writeln
end-do


end-model
```

<u>Question2: Solution Report:</u>

Objective: Minimise Maximum of all Overtimes = 27.2121

Decision variables:

	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Overtime (hrs)	27.2121	27.2121	27.2121	27.1861	27.2051

Production Schedule	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Part 1	1308	2192	0	0	0
Part 2	0	341	2264	395	0
Part 3	0	0	0	4000	0
Part 4	2991	0	0	0	1009
Part 5	0	0	0	0	2800

Question 3

XPRESS Implementation:

```
!@encoding CP1252
model FalconDieCasting
uses "mmxprs"; !gain access to the Xpress-Optimizer solver

declarations
	NUMMACHINES = 5
	NUMPARTS = 5
	MACHINERANGE = 1..NUMMACHINES		!j
	PARTRANGE = 1..NUMPARTS		!i
	u: real !maximum regular time per week
	v: real !maximum overtime per week
	c1: real !cost of overtime per hour per machine (related to Overtime Sum)
	c2: real !cost of overtime per hour that at least one machine is on overtime (related to Max Overtime or Largest Hours

	b: array(PARTRANGE) of real !demand for each part
	s: array(PARTRANGE, MACHINERANGE) of real !setup time for producing each part on each machine
	y: array(PARTRANGE) of real !percentage yield of useable parts for each part
	r: array(PARTRANGE, MACHINERANGE) of real !production rate of each part on each machine, needs to be initialised
and then * by yield

	M: array(PARTRANGE, MACHINERANGE) of real !big M, max amount P could be

	!DECISION VARIABLES
	P: array(PARTRANGE,MACHINERANGE) of mpvar !amount produced of each part on each machine
	X: array(PARTRANGE, MACHINERANGE) of mpvar !binary, machine used to make part
	O: array(MACHINERANGE) of mpvar	!Overtime for each machine

	!OBJECTIVE FUNCTION
	LargestHours:mpvar !Question 2: Largest number of hours for one machine
	OvertimeCost: linctr !Question 3: Cost for overtime

	!CONSTRAINTS
	TimeCS: array(MACHINERANGE) of linctr
	DemandCS: array(PARTRANGE) of linctr
	MCS: array(PARTRANGE, MACHINERANGE) of linctr
	OCS: array(MACHINERANGE) of linctr
	OMCS: array(MACHINERANGE) of linctr

end-declarations
	!initialise all constants/variables
	u := 120.0 !maximum regular time per week
	v := 48.0 !maximum overtime per week

	c1 := 30 !cost of scheduling overtime per machine per hour of overtime
	c2 := 40 !cost of support personnel, per hour of overtime

	b :: [	3500,	3000,	4000,	4000,	2800] !demand for each part

	s :: [	8,		10,		999,	999,	999,!setup time for producing each part on each
machine
			999,	8,		10,		8,		999,
			999,	999,	999,	12,		999,
			8,		999,	999,	999,	8,
			999,	999,	24,		999,	20]
```

```
	r :: [	40.0,	35.0,	0.0,	0.0,	0.0, !production rate initial values
			0.0,	25.0,	30.0,	35.0,	0.0,
			0.0,	0.0,	0.0,	50.0,	0.0,
			60.0,	0.0,	0.0,	0.0,	60.0,
			0.0,	0.0,	45.0,	0.0,	50.0]

	y :: [0.6,	0.55, 0.75, 0.65, 0.6]

	!initial calculations, set to binary or integer
	forall(j1 in 1..NUMMACHINES) do
		forall(i1 in 1..NUMPARTS) do
			P(i1,j1) is_integer	!make P, amount produced an integer vlaue
			X(i1,j1) is_binary	!make X, whether or not started
			r(i1,j1) := r(i1,j1)*y(i1)		!calculate r, taking into account yield of each machines
			M(i1,j1) := r(i1,j1)*(u + v - s(i1,j1))
		end-do
	end-do

	!OBJECTIVE FUNCTION QUESTION 1
	OvertimeSumOF := sum(j2 in 1..NUMMACHINES) O(j2)

	!TIME CONSTRAINTS
	forall(j3 in 1..NUMMACHINES) do
		!timetaken(j3) := sum(i3 in 1..NUMPARTS) (P(i3,j3)/r(i3,j3) + X(i3,j3)*s(i3,j3))
		TimeCS(j3) := sum(i3 in PARTRANGE | r(i3,j3)>0) (P(i3,j3)/r(i3,j3) + X(i3,j3)*s(i3,j3)) <=  u + O(j3)
		!writeln("Time Constraint(",j3, ") is:", TimeCS(j3) )
	end-do

	!DEMAND CONSTRAINTS
	forall(i4 in 1..NUMPARTS) do
		!DemandCS(i4) :=	sum(j4 in 1..NUMMACHINES) P(i4,j4) >= b(1, i4) !demand CS for 1st week
		DemandCS(i4) :=	(sum(j4 in 1..NUMMACHINES) P(i4,j4)) >= b(i4) !demand CS for 2nd week
	end-do

	!BIG M CONSTRAINTS
	forall(i5 in PARTRANGE) do
		forall(j5 in MACHINERANGE) do
			MCS(i5,j5) := P(i5,j5) <= M(i5,j5)*X(i5,j5)
		end-do
	end-do

	!MAX OVERTIME CONSTRAINT
	forall(j6 in MACHINERANGE) do
		OCS(j6) := O(j6) <= v
	end-do

	!MAX MACHINE OVERTIME CONSTRAINT Objective for Question 2
	forall(j7 in MACHINERANGE) do
		OMCS(j7) := O(j7) <= LargestHours
	end-do

	!OBJECTIVE FUNCTION QUESTION 3
	OvertimeCost := c1*(sum(j8 in 1..NUMMACHINES) O(j8)) + c2*LargestHours

minimise(OvertimeCost) !Question 3
writeln("Question 3 Objective is Cost of Overtime = ", getsol(OvertimeCost))
```

```
write("Overtime hours: ")
forall(j8 in MACHINERANGE) do
        write(getsol(O(j8)))
        write(", ")
end-do
writeln
forall (i in 1..NUMMACHINES) do
        forall (j in 1..NUMPARTS) do
                write(getsol(P(i,j)))
                write(", ")
        end-do
        writeln
end-do
end-model
```

<u>Question 3: Solution Report:</u>

Objective Minimise the overtime costs: 5158.36

	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Overtime (hrs)	33.9167	0	36.7273	36.7446	15.5641

Production Schedule	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Part 1	1190	2310	0	0	0
Part 2	0	0	2421	579	0
Part 3	0	0	0	4000	0
Part 4	3445	0	0	0	555
Part 5	0	0	0	0	2800

Question 4

XPRESS Implementation

```
!@encoding CP1252
model FalconDieCasting
uses "mmxprs"; !gain access to the Xpress-Optimizer solver

declarations
   NUMMACHINES = 5
   NUMPARTS = 5
   NUMWEEKS = 2
   MACHINERANGE = 1..NUMMACHINES     !j
   PARTRANGE = 1..NUMPARTS    !i
   WEEKRANGE = 1..NUMWEEKS
   u: real !maximum regular time per week
   v: real !maximum overtime per week
   c1: real !cost of overtime per hour per machine (related to Overtime Sum)
   c2: real !cost of overtime per hour that at least one machine is on overtime (related to Max Overtime or Largest Hours

   b: array(WEEKRANGE, PARTRANGE) of real !demand for each part
   s: array(PARTRANGE, MACHINERANGE) of real !setup time for producing each part on each machine
   y: array(PARTRANGE) of real !percentage yield of useable parts for each part
   r: array(PARTRANGE, MACHINERANGE) of real !production rate of each part on each machine, needs to be initialised and then
* by yield

   M: array(PARTRANGE, MACHINERANGE) of real !big M, max amount P could be

   !DECISION VARIABLES
   P: array(PARTRANGE,MACHINERANGE) of mpvar !amount produced of each part on each machine
   X: array(PARTRANGE, MACHINERANGE) of mpvar !binary, machine used to make part
   O: array(MACHINERANGE) of mpvar    !Overtime for each machine
   U: array(PARTRANGE) of mpvar !Unsatisfied demand for each part

   !OBJECTIVE FUNCTION
   UnsatisfiedDemandCost: linctr !Question 4: Sum of unsatisfied demand, which needs to be minimised
   LargestHours:mpvar !Question 2: Largest number of hours for one machine

   !CONSTRAINTS
   TimeCS: array(MACHINERANGE) of linctr
   DemandCS: array(PARTRANGE) of linctr
   MCS: array(PARTRANGE, MACHINERANGE) of linctr
   OCS: array(MACHINERANGE) of linctr
   OMCS: array(MACHINERANGE) of linctr
end-declarations
   !initialise all constants/variables
   u := 120.0 !maximum regular time per week
   v := 48.0 !maximum overtime per week

   c1 := 30 !cost of scheduling overtime per machine per hour of overtime
   c2 := 40 !cost of support personnel, per hour of overtime

   b :: [    3500,    3000,    4000,    4000,    2800, !demand for each part
                       4500,    4000,    5000,    5000,    3800]
          s :: [    8,                 10,          999,    999,    999,!setup time for producing each part on each machine
                                 999,    8,                 10,          8,             999,
                                 999,    999,    999,    12,          999,
                                 8,            999,    999,    999,    8,
```

```
				999,	999,	24,	999,	20]

		r :: [	40.0,	35.0,	0.0,	0.0,	0.0, !production rate initial values
				0.0,	25.0,	30.0,	35.0,	0.0,
				0.0,	0.0,	0.0,	50.0,	0.0,
				60.0,	0.0,	0.0,	0.0,	60.0,
				0.0,	0.0,	45.0,	0.0,	50.0]

		y :: [0.6,	0.55, 0.75, 0.65, 0.6]

	!initial calculations, set to binary or integer
	forall(j1 in 1..NUMMACHINES) do
		forall(i1 in 1..NUMPARTS) do
			P(i1,j1) is_integer	!make P, amount produced an integer vlaue
			X(i1,j1) is_binary	!make X, whether or not started
			r(i1,j1) := r(i1,j1)*y(i1)	!calculate r, taking into account yield of each machines
			M(i1,j1) := r(i1,j1)*(u + v - s(i1,j1))
			U(i1) is_integer
		end-do
	end-do

	!TIME CONSTRAINTS
	forall(j3 in 1..NUMMACHINES) do
		TimeCS(j3) := sum(i3 in PARTRANGE | r(i3,j3)>0) (P(i3,j3)/r(i3,j3) + X(i3,j3)*s(i3,j3)) <=  u + O(j3)
	end-do

	!DEMAND CONSTRAINTS
	forall(i4 in 1..NUMPARTS) do
		DemandCS(i4) :=	(sum(j4 in 1..NUMMACHINES) P(i4,j4)) + U(i4) >= b(2, i4) !demand CS for 2nd week
	end-do

	!BIG M CONSTRAINTS
	forall(i5 in PARTRANGE) do
		forall(j5 in MACHINERANGE) do
			MCS(i5,j5) := P(i5,j5) <= M(i5,j5)*X(i5,j5)
		end-do
	end-do

	!MAX OVERTIME CONSTRAINT
	forall(j6 in MACHINERANGE) do
		OCS(j6) := O(j6) <= v
	end-do

	!MAX MACHINE OVERTIME CONSTRAINT Objective for Question 2
	forall(j7 in MACHINERANGE) do
		OMCS(j7) := O(j7) <= LargestHours
	end-do

	!OBJECTIVE FUNCTION QUESTION 4
	UnsatisfiedDemandCost := (sum(i9 in PARTRANGE) U(i9))*3 + (sum(j in 1..NUMMACHINES) O(j))*c1 + LargestHours*c2


minimise(UnsatisfiedDemandCost) !Question 4
writeln("Question 4 Objective is Cost of Unsatisfied Demand = ", getsol(UnsatisfiedDemandSum))
write("Overtime hours: ")
forall(j8 in MACHINERANGE) do
	write(getsol(O(j8)))
```

```
   write(", ")
end-do
writeln
forall (i in 1..NUMMACHINES) do
   forall (j in 1..NUMPARTS) do
            write(getsol(P(i,j)))
            write(", ")
   end-do
   writeln
end-do
end-model
```

Question 4: Solution Report:

	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Overtime (hrs)	47.9968	48	48	47.9827	48

Production Schedule	Machine 1	Machine 2	Machine 3	Machine 4	Machine 5
Part 1	891	3318	0	0	0
Part 2	0	0	2607	282	0
Part 3	0	0	0	5000	0
Part 4	4480	0	0	0	520
Part 5	0	0	0	0	3800

U_i: Potential (in this case actual) Shortfalls

	Part 1	Part 2	Part 3	Part 4	Part 5
Unsatisfied Demand	291	1111	0	0	0