

# ILOG CPLEX OPL Modelling for Machine Cell Formation

Subhaa R.<sup>1</sup>, Dr.N.Jawahar<sup>2</sup>,

<sup>1</sup>Research Scholar, <sup>2</sup>Professor of Mechanical Engineering & Dean (R&D),  
Department of Mechanical Engineering,  
Thiagarajar College of Engineering,  
Madurai – 625 015, Tamilnadu, India.

<sup>1</sup>[rsubhaa@yahoo.com](mailto:rsubhaa@yahoo.com)

<sup>2</sup>[jawahartce@tce.edu](mailto:jawahartce@tce.edu), [jawahartce@yahoo.co.uk](mailto:jawahartce@yahoo.co.uk)

**Abstract**— This paper proposes CPLEX solver for the Cell Formation (CF) problem of the cellular manufacturing system. The CF problem deals with the assignment of parts and machines to cell to optimize the required objectives. A mathematical model is developed for the CF model with the objective to minimize the sum of the costs of scheduling, cross-flow, inter-cell movement and machine duplication. The mathematical model is coded in CPLEX Optimisation Programming Language (OPL), a mathematical programming language of IBM ILOG CPLEX optimization studio and illustrated with a literature problem. The CPLEX solver provides optimal solution to the CF problem in a reasonable computational time.

**Keyword-** Cellular manufacturing, scheduling, mathematical model, cross-flow, optimization, CPLEX solver.

## I. INTRODUCTION

IBM ILOG CPLEX Optimization Studio is an analytical decision support toolkit for rapid development and deployment of optimization models using mathematical and constraint programming. It combines an integrated development environment (IDE) with the powerful Optimization Programming Language (OPL) and high-performance ILOG CPLEX optimizer solvers. The CPLEX CP optimiser component is a second-generation constraint programming engine for automatically solving detailed scheduling problems[1].

CPLEX solver is used to mathematically program and solve the engineering problems like inventory [2], capacity planning [3], scheduling [4] and facility location [5]. In this paper, the Cell Formation (CF) problem of the Cellular Manufacturing System (CMS) is modelled and solved using CPLEX solver. The CPLEX solver is also used in papers [6], [7], [8] for solving CF problems. The CF problem, an important design stage of CMS, deals with the grouping of parts and their required machines to optimise one or more objectives.

During cell formation, it is not possible in most of the industrial cases to group independent cells, which are the cells with no inter-cell movement and a perfect grouping of parts and machines prevails. The inter-cell movement is the movement of parts from parent cell to another cell for the requirement of certain operations of the parts being done in the machine available only in another cell. Such movement incurs a cost called inter-cell movement cost (ICC). Thus, to avoid ICC, machines are duplicated in the possible cases, which incurs a cost called machine duplication cost (MDC). The machine duplications apart from reducing ICC, it also reduces makespan time (i.e. time to complete all the parts) due to increased machine resource availability. This makespan time is modelled as a cost term called scheduling cost (SCT). Thus reduction in ICC reduces makespan time and in turn reduces SCT. Further, machine duplications enable cross-flow, a term used to indicate movement of parts from parent cell to another cell in spite of availability of machine in the parent cell with the objective to reduce makespan time and thus reduce SCT. This type of movement incurs a cost named as cross-flow movement cost (CFC).

With these considerations of cost evaluation parameters, the CF problem is modelled with the objective to minimise total cost (TC) that includes costs of SCT, CFC, ICC and MDC. The proposed CF model is mathematically formulated and solved using OPL syntax provided by IBM ILOG CPLEX optimisation studio. The solution procedure is discussed with a literature problem.

The rest of the paper is organised with problem description in section II. The section III provides the CPLEX solution procedure and numerical illustration is given in section IV. The section V gives the conclusion and future research directions.

## II. PROBLEM DESCRIPTION

### A. Problem Environment

The CF problem of the CMS needs to process 'n' parts (indexed as 'i') with each part 'i' has a specific processing sequence consisting of ' $K_i$ ' number of operations (indexed as 'k'). The parts require 'm' number of machine types (referred as machine and indexed as 'j') to process all their operations. The machine ' $j_{ik}$ ' required for the  $k^{\text{th}}$  operation of part 'i' (represented as ' $O_{ik}$ ') and its unit processing time ' $t_{ik}$ ' are known for all operations (i.e.  $\forall i = 1$  to  $n$  and  $\forall k = 1$  to  $K_i$ ). The unit processing time  $t_{ik}$  includes set-up time of operation  $O_{ik}$ . The parts are non-preemptive and are available at time zero. The demand  $d_i$  for each part 'i' for a planning period is known. The parts and the machines are to be grouped into exactly 'C' number of cells (indexed as 'c'). Since the machines required to carry all the operations  $O_{ik}$  are available in any one of the cell 'c', the parts are not subcontracted and entire demand is met with in the CMS. Further, the demands of parts are to be processed as a whole (i.e. lot splitting is not considered). The duplication of machine 'j' within the cell is not allowed and hence, the maximum number of any machine 'j' in the CMS, thus becomes equal to the number of cells 'C'. It is assumed that the machines duplicated are identical to the existing machines and hence the processing time is same for the operations that are to be processed in duplicated machines. This model allows both inter-cell moves and cross-flow moves of parts.

### B. Problem Statement

The problem is to determine the optimal CF along with makespan schedule for minimum total cost of scheduling, cross-flow movement, inter-cell movement and machine duplication given:

- Parts related data: number of parts 'n' and number of machines 'm', part processing characteristics of operation sequence ' $K_i$ ', machines required ' $j_{ik}$ ', unit processing time ' $t_{ik}$ ' and demand ' $d_i$ ',
- Cost coefficients related data: scheduling cost per unit time (SC), machine duplication cost of machine 'j' ( $MD_j$ ), inter-cell movement cost  $IC_{c'c}$  and cross-flow cost  $CF_{c'c}$  for per unit part and
- Number of cells 'C'

### C. Mathematical Formulation

Based on the model delineated in the previous section, the mathematical formulation is done and is given as follows:

#### Nomenclature

#### Indices

- c, c' index for cells (c, c' = 1, 2, ..., C)
- i, p index for parts (i = 1, 2, ..., n)
- j, q index for machines (j = 1, 2, ..., m)
- k index for processing sequence number (k = 1, 2, ...,  $K_i$ )

#### Model Parameters

- C number of cells
- $C_{iK_i}$  completion time of the last  $K_i^{\text{th}}$  operation  $O_{iK_i}$  of part 'i'
- $C_{ik}$  completion time of the operation  $O_{ik}$  of part 'i'
- $CF_{c'c}$  cross-flow movement cost per unit part from cell 'c' to cell 'c'
- $d_i$  demand for part 'i' per planning period
- $IC_{c'c}$  inter-cell movement cost per unit part from cell 'c' to cell 'c'
- $j_{ik}$  machine required for the operation  $O_{ik}$
- n number of parts
- m number of machines
- $MD_j$  machine duplication cost of machine 'j' per planning period
- $O_{ik}$   $k^{\text{th}}$  operation of part 'i'
- SC scheduling cost for unit time
- $S_{ik}$  start time of operation  $O_{ik}$
- $t_{ik}$  processing time for  $k^{\text{th}}$  operation sequence of part 'i'
- TC total cost of operation

#### Model Variables

- $X_{j_{ik}c}$  binary integer variable that indicates the assignment of machine 'j' to cell c

- $Y_{jik.c}$  binary integer variable that indicates operation  $O_{ik}$  is assigned to machine 'j' available in cell c
- $Z_{i.c}$  binary integer variable that indicates the assignment of part 'i' to cell c
- $\alpha_{ikpq.c}$  binary integer variable that indicates the precedence relationship of operations  $O_{ik}$  and  $O_{pq}$  assigned in cell 'c'

$$\begin{aligned} \text{Minimize TC} &= [Max(C_{1K_1} \dots \dots C_{nK_n})] \times (SC) \\ &+ \left[ \sum_{i=1}^n \sum_{k=1}^{K_i} \sum_{c'=1}^C \sum_{c=1}^C CF_{c'c} \times d_i \times [(1 - Y_{jik.c}) \times X_{jik.c} \times Z_{i.c}] \right] \\ &+ \left[ \sum_{i=1}^n \sum_{k=1}^{K_i} \sum_{c'=1}^C \sum_{c=1}^C IC_{c'c} \times d_i \times [(1 - Y_{jik.c}) \times (1 - X_{jik.c}) \times Z_{i.c}] \right] \\ &+ [(\sum_{c=1}^C \sum_{j=1}^m X_{jik.c} \times MD_j) - (\sum_{j=1}^m MD_j)] \end{aligned} \tag{1}$$

Subject to:

$$\sum_{c=1}^C Z_{i.c} = 1 \quad \forall i \tag{2}$$

$$\sum_{i=1}^n Z_{i.c} \geq 1 \quad \forall c \tag{3}$$

$$\sum_{c=1}^C X_{jik.c} \geq 1 \quad \forall i, k \tag{4}$$

$$\sum_{c=1}^C Y_{jik.c} = 1 \quad \forall i, k \tag{5}$$

$$Y_{jik.c} \leq X_{jik.c} \quad \forall i, k, c \tag{6}$$

$$S_{ik} \geq 0 \quad \forall i, k \tag{7}$$

$$C_{ik} - S_{ik} - (d_i \times t_{ik}) = 0 \quad \forall i, k \tag{8}$$

$$S_{i(k+1)} - C_{ik} \geq 0 \quad \forall i, k = 1 \dots (K_i - 1) \tag{9}$$

$$\left( (Y_{jik.c} \times C_{ik}) - (Y_{jpq.c} \times C_{pq}) \right) \alpha_{ikpq.c} - \left( (Y_{jik.c} \times C_{ik}) - (Y_{jpq.c} \times C_{pq}) \right) (1 - \alpha_{ikpq.c}) \geq \alpha_{ikpq.c} (d_p \times t_{pq}) + (1 - \alpha_{ikpq.c})(d_p \times t_{pq}) \quad \forall c, (i, k), (p, q), i \neq p, j_{pq} = j_{ik} \tag{10}$$

$$\left( (Y_{jik.c} \times C_{ik}) - (Y_{jpq.c} \times C_{pq}) \right) \alpha_{ikpq.c} \geq 0 \quad \forall c, (i, k), (p, q), i \neq p, j_{pq} = j_{ik} \tag{11}$$

$$X_{jik.c}, Y_{jik.c}, Z_{i.c} \in \{0,1\} \quad \forall i, k, c \tag{12}$$

$$\alpha_{ikpq.c} \in \{0,1\} \quad \forall (i, k), (p, q), c \tag{13}$$

Where

$$\alpha_{ikpq.c} = \begin{cases} 1 & \text{if } (Y_{jik.c} \times C_{ik}) - (Y_{jpq.c} \times C_{pq}) > 0 \text{ or if } O_{pq} \text{ precedes } O_{ik} \\ 0 & \text{otherwise} \end{cases}$$

$$X_{jik.c} = \begin{cases} 1 & \text{if machine 'j' is assigned to cell 'c'} \\ 0 & \text{otherwise} \end{cases}$$

$$Y_{jik.c} = \begin{cases} 1 & \text{if operation } O_{ik} \text{ is assigned to machine 'j' available in cell 'c'} \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{i.c} = \begin{cases} 1 & \text{if part 'i' is assigned to cell 'c'} \\ 0 & \text{otherwise} \end{cases}$$

The objective of the problem is to minimize the total cost of operation (TC) and is given as equation (1). The first term of the objective function gives the scheduling cost (SCT), the second term is cross-flow movement cost (CFC), the third is inter-cell movement cost (ICC) and the fourth being machine duplication cost (MDC). The constraint (2) ensures each part 'i' is assigned to only one cell. The constraint (3) assures that at least one part is assigned to a cell. The constraint (4) ensures that at least one machine 'j' is available in any one of the cells. The constraints (5)-(11) are related to the scheduling of the part operations in CMS. The constraint (5) satisfies the condition that each operation should be assigned to only one cell. The constraint (6) ensures that an operation  $O_{ik}$  is assigned to the cell, only when the machine  $j_{ik}$  required for that operation is available in that cell and further it checks the availability of cross-flow for each part operation. The constraint (7) ensures that start time of an operation ( $O_{ik}$ ) is always positive. The condition that the parts are non-pre-emptive is assured in the constraint (8). In this constraint, the difference between the completion time ' $C_{ik}$ ' and the starting time ' $S_{ik}$ '

of an operation ' $O_{ik}$ ' is equal to its processing time ' $d_i \times t_{ik}$ '. The constraint (9) ensures precedence relationship among operations of a part. The constraints (10) and (11) control that no two operations are processed simultaneously on the same machine. In the constraint (10), the difference between the completion times of two operations  $O_{ik}$  and  $O_{pq}$  ( $O_{pq}$  precedes  $O_{ik}$ ) that requires the same machine 'j' available in cell 'c' should be greater than or equal to total operation time of the later operation. The constraint (11) assures the precedence relationship among operations done on the machine 'j' available in cell 'c'. The constraint (12) and (13) represents the binary integer requirements of the decision variables and control variable.

### III. PROPOSED SOLUTION METHODOLOGY

#### D. Procedure of CPLEX Solver

The CF decisions of part and machine cell assignment and decisions of part operations assignment with alternatives as cross-flow are 0-1 integer programming and belong to combinatorial optimization problem. The combinatorial explosion occurs with the increase in number of machines to be duplicated and thus corresponding increase in the number of available cross-flows for the part operations. The characteristics of the model require an efficient solution procedure. CPLEX, a feature of IBM ILOG Optimization Studio, offers state of the art performance and robustness in an optimization engine for solving problems expressed as mathematical programming models. CP Optimizer, also a feature of IBM ILOG Optimization Studio, is a software library of constraint programming tools supporting constraint propagation, domain reduction, and highly optimized solution search [9]. IBM ILOG CP optimizer has been used for job-shop scheduling and shown as a proven tool for small and medium size problems [10]. On this concern, this paper proposes a procedure developed using the mathematical language of CPLEX solver for solving the mathematical formulation presented in section II-C. The CPLEX CP optimizer program codes and data input are given in appendix I and II.

### IV. NUMERICAL ILLUSTRATION

The proposed solution procedure is illustrated with a literature problem [11] of 7 parts (n) that requires processing by 6 machine-types (m) in 2 cells (C). The machines-type required ' $j_{ik}$ ' for each part 'i' are adopted from literature problem. The other parts related data for each part 'i' and its operation 'k' ( $\in K_i$ ) are generated randomly as delineated below:

- Processing time ' $t_{ik}$ ': Uniform distribution in the range of 2-10 unit times.
- Demand ' $d_i$ ': Uniform distribution in the range of 100-250.

Table I provides the parts related data generated for the illustrative problem. The cost related data have been assumed as follows to address reasonably the practical manufacturing scenario:

- Scheduling cost per unit time (SC), based on overhead charges for the manufacturing as 1 per unit time
- Machine duplication cost ( $MD_j$ ) of machine 'j' per planning period is fixed on the basis of depreciation value and generated randomly using uniform distribution in the range of 500-1500 and is given in Table II.
- Cross-flow cost ( $CF_{c'c}$ ) and inter-cell movement cost ( $IC_{c'c}$ ) are considered to be same on the assumption that movement of parts within the cell (i.e) intra-cell movement is negligible and cost of moving the parts from cell 1 to cell 2 and vice-versa are the same. The cost matrix for the example problem with 2 cells and assuming the CMS cell layout as linear is given in Table III.

TABLE I  
Processing Characteristics of Illustrative Problem

		Operation No. (k)			Demand (d <sub>i</sub> )
		1	2	3	
1	j <sub>ik</sub>	3	5	1	100
	t <sub>ik</sub>	7	4	10	
2	j <sub>ik</sub>	6			250
	t <sub>ik</sub>	5			
3	j <sub>ik</sub>	3	1	4	100
	t <sub>ik</sub>	10	5	3	
4	j <sub>ik</sub>	1	4		160
	t <sub>ik</sub>	5	10		
5	j <sub>ik</sub>	5			150
	t <sub>ik</sub>	6			
6	j <sub>ik</sub>	5	6	2	100
	t <sub>ik</sub>	10	5	8	
7	j <sub>ik</sub>	5	2		200
	t <sub>ik</sub>	10	5		

TABLE II  
Machine Duplication Cost (MD<sub>i</sub>)

Machine j	1	2	3	4	5	6
Machine duplication cost MD <sub>i</sub>	600	500	1000	200	500	1200

TABLE III  
Inter-cell and Cross-flow cost matrix for unit part movement

To	Cell 1	Cell 2
From Cell 1	0	1
From Cell 2	1	0

The illustrative problem is run on HP dx2280 MT model X86 based desktop computer with the CPLEX solver version 12.4. The data are given as input to the CPLEX program. The cell formation details indicating parts and machine grouping in cells 1 and 2 for the illustrative problem are given in Table IV. It is noted from Table IV, that the machine 5 is duplicated once in this cell formation configuration. The cell formation details in matrix form are given in Table V to show cross-flow movement. The Table V reveals that the part 6 has an inter-cellular movement from cell 1 to machine 5 in cell 2. This movement is termed as cross-flow movement of parts as even though machine 5, available in cell 1, the part operation of part 6 with the parent cell as cell 1, is assigned to cell 2. Fig. 1. and Fig. 2. show the optimal schedule as Gantt chart for part-wise and machine-wise minimum makespan production schedule respectively generated by CPLEX solver. It is seen from the Fig. 2. that the 1<sup>st</sup> operation of part 6 (i.e. Ops[<6,1>][2] shown in Fig. 1. is assigned to cell 2, which if not allowed, will make the makespan time to increase, which thus increases scheduling cost and thus the total cost of operation. The objective function value for the cell formation with minimum cost of operation is given in Table VI with the various cost parameters of SCT, ICC, CFC and MDC and with the details of their values in remarks column.

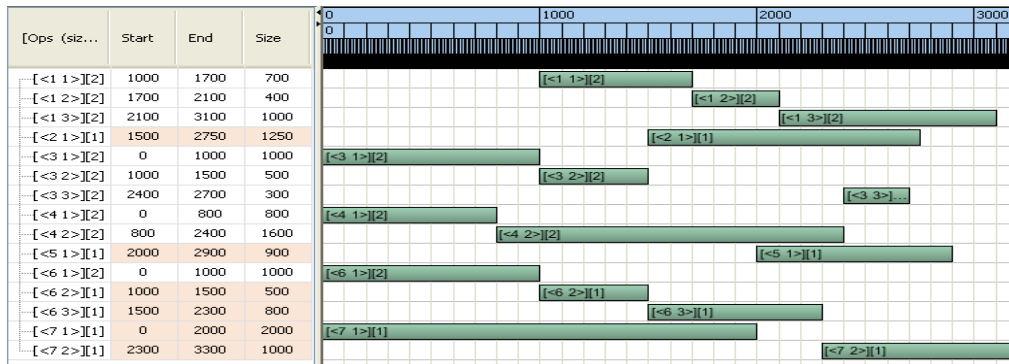
TABLE IV  
Cell Formation Details for Illustrative Problem

Cell no.	Parts	Machines
1	1,3,4,7	1,3,4, <u>5</u>
2	2,5,6	2, <u>5</u> ,6

TABLE V  
Cell Formation Matrix for Illustrative Problem

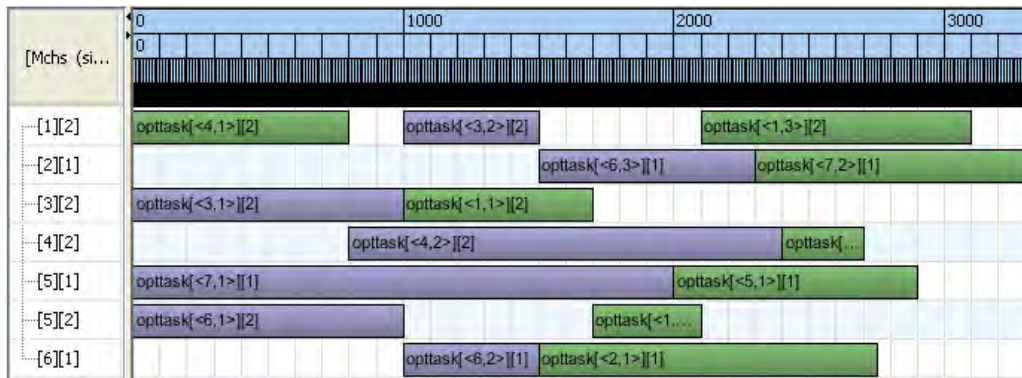
		Machine (j)						
		Cell 1			Cell 2			
Part (i)		2	5	6	1	3	4	5
2	Cell 1			1				
5			1					
6			1					1
7		1	1					
1	Cell 2				1	1		1
3					1	1	1	
4					1		1	

Note: striped element indicates the cross-flow movement



Note : Column 1 Ops[<4,1>][2] means 1<sup>st</sup> operation of part 4 is assigned to cell 2. Column 2,3,4 gives schedule of the operations

Fig. 1. Gantt chart (part-wise) for optimal schedule generated using CPLEX solver



Note : Mchs[5][1] means Machine-type 5 assigned to cell 1.

Fig. 2 : Gantt chart (Machine-wise) for optimal schedule generated using CPLEX solver

TABLE VI  
Cost Parameters for Illustrative Problem

Cost parameter	Cost	Remarks
Machine duplication cost (MDC)	500	Machines 5 is duplicated once
Inter-cellular movement cost (ICC)	0	No inter-cellular movement
Cross-flow cost (CFC)	100	Part 6 has Cross-flow movement (100x1)
Scheduling cost (SCT)	3300	Makespan time x SC (3300X1)
Total Cost (TC)	3900	TC=MDC+ICC+CFC+SCT

V. CONCLUSION

In this paper, the CF problem of CMS based on scheduling of part operations and cross-flow is addressed considering machine duplications. A mathematical model is developed to evolve the optimal machine cell grouping and optimal scheduling of part operations with cross-flow enabled due to machine duplications. The

model is solved using CPLEX solver version 12.4 and illustrated. The solver is able to find optimal solutions for small and medium size problems. With respect to the capability of the solver, it is not able to find an optimal solution within a reasonable computational time for larger- size of problems. Hence, in future research meta-heuristics like simulated annealing, genetic algorithm can be used for part and machine assignment decisions as well as operation assignment decisions.

#### ACKNOWLEDGMENT

This research is supported by WOS-A Scheme of Department of Science and Technology (DST) functioning under the Ministry of Science and Technology, India. The authors thank the Management of Thiagarajar College of Engineering, Madurai, for the facilities provided to carry out this research and thank DST for the financial support.

#### REFERENCES

- [1] IBM ILOG website [Online]. Available: <http://www-01.ibm.com/software/in/websphere/products/optimisation/index.html>.
- [2] Mayur Vamanan, Qian Wang, Rajan Batta, and Robert J. Szczerba, "Integration of COTS software products ARENA & CPLEX for an inventory/logistics problem," *Computers & Operations Research*, 31 533–547, 2004.
- [3] R. Manerba, Mansini, "An effective metaheuristic for the capacitated total quantity discount problem," *Computers & Operations Research*, 41, 1–11, 2014
- [4] M. Karimi-Nasab, S.M.Seyedhoseini, "Multi-level lot sizing and job shop scheduling with compressible process times: A cutting plane approach," *European Journal of Operational Research*, 231, 598–616, 2013.
- [5] Amara André R.S., "A parallel ordering problem in facilities layout," *Computers & Operations Research*, 40, 2930–2939, 2013.
- [6] Steve Ah kioon, Akif Asil Bulgak, Tolga Bektas, "Integrated cellular manufacturing systems design with production planning and dynamic system reconfiguration," *European Journal of Operational Research*, 192 , 414–428, 2009.
- [7] Poornachandra Rao Panchalavarapu, and Vira Chankong, "Design of cellular manufacturing system with assembly considerations," *Computers & Industrial Engineering*, Volume 48, Issue 3, 449-469, 2005.
- [8] Mingyuan Chen, and Dong Cao, "Coordinating production planning in cellular manufacturing environment using Tabu search," *Computers & Industrial Engineering*, Volume 46, Issue 3, pg 571-588, 2004
- [9] IBM ILOG CPLEX Optimisation Studio V12.4 – Reference manual
- [10] B. S. Girish, and N. Jawahar, 'Scheduling job shop associated with multiple routings with genetic and ant colony heuristics', *International Journal of Production Research*, 47(14), 3891-3917, 2009.
- [11] K.L. Mak, Y.S. Wong, and X.X. Wang, "An Adaptive Genetic Algorithm for Manufacturing Cell Formation," *International Journal of Advanced Manufacturing Technology*, 16, 491–497, 2000.

#### APPENDIX I

##### CPLEX Program Code

using CP;

int nbparts=...; int nbcells=...; int nbmchs=...;

range parts=1..nbparts; range cells=1..nbcells; range Mchs=1..nbmchs;

//input parameters

int dem[parts]=...; int mcos[Mchs]=...; int Movcos[cells,cells]=...; int CF\_cos[cells,cells]=...; int scos=...;

```
tuple Operation {
  key int jobId; // Part id
  key int pos; // Position in job
  int mch; // Machine
  int pt; // Processing time
};
```

```
{Operation} Ops = ...;
// Position of last operation of job j
int jlast[j in parts] = max(o in Ops: o.jobId==j) o.pos;
```

```
dvar boolean partassign[parts,cells];
dvar boolean mcassign[Mchs,cells];
dvar interval opseq[md in Ops] size md.pt;
dvar interval opttask[o in Ops][c in cells] optional size o.pt;
dvar sequence mchs[m in Mchs][c in cells] in all(md in Ops: md.mch==m) opttask[md][c];
```

```
dexpr int mkspancos=max(c in cells,j in parts, o in Ops: o.pos==jlast[j])
endOf(opttask[o][c])*scos*presenceOf(opttask[o][c]);
dexpr int CFcos=(sum(c1 in cells,c2 in cells,o in
Ops)CF_cos[c1,c2]*dem[o.jobId]*presenceOf(opttask[o,c2])*partassign[o.jobId,c1]*mcassign[o.mch,c1]);
```

```

dexpr int ICcos=(sum(c1 in cells,c2 in cells,o in
Ops)Movcos[c1,c2]*dem[o.jobId]*presenceOf(opttask[o,c2])*partassign[o.jobId,c1]*(1-mcassign[o.mch,c1]));
dexpr int MDcos=((sum(c in cells,j in Mchs)mcassign[j,c]*mcos[j])-(sum(j in chs)mcos[j]));
minimize mkspancos+CFcos+ICcos+MDcos;
subject to
{
//assure each part is assigned to only one cell - Constraint 2
forall(i in parts) sum(c in cells) partassign[i,c]==1;

//atleast one part is assigned to a cell - constraint 3
forall(c in cells) sum(i in parts) partassign[i,c]>=1;

//ensures atleast one machine 'j' available in anyone of the cells - constraint 4
forall(j in Mchs) sum(c in cells) mcassign[j,c]>=1;

//ensures operations are assigned to only one cell - constraint 5
forall(i in parts,j in Mchs) sum(c in cells) opassign[i,j,c]==1;

//part operation assignment and cross-flow control assignment - constraint 6
forall(o in Ops,c in cells) presenceOf(opttask[o,c])<= mcassign[o.mch,c];

//part operation precedence check - constraint 10
forall (i in parts, o1 in Ops, o2 in Ops: o1.jobId==i && o2.jobId==i && o2.pos==1+o1.pos)
endBeforeStart(opseq[o1],opseq[o2]);

//cross-flow
forall (o in Ops)
alternative(opseq[o],all(c in cells)opttask[o][c]);

//machine overlap check
forall (m in Mchs,c in cells) noOverlap(mchs[m][c]);
}
}

```

## APPENDIX II

## Input Data for CPLEX program

```

nbcells=2;
nbparts=7;
nbmchs=6;

dem=[200 250 100 160 150 225 200];
mcos=[600 500 1000 200 500 1200];
Movcos=[[0,1],
[1,0]];
CF_cos=[[0,1],
[1,0]];
scos=10;

Ops={<1,1,3,1400>, <1,2,5,800>, <1,3,1,2000>,
<2,1,6,1250>,
<3,1,3,1000>, <3,2,1,500>, <3,3,4,300>,
<4,1,1,800>, <4,2,4,1600>,
<5,1,5,900>,
<6,1,5,2250>, <6,2,6,1125>, <6,3,2,1800>,
<7,1,5,2000>, <7,2,2,1000>
};

```