

```

! Flow Shop Scheduling with Change_Over Times;
! In a flow shop, each job gets processed once, in order, on machines: 1, 2, 3, ...
! There is a changeover time, e.g., cleaning, between successive jobs on a machine.
! The objective is to minimize the maximum finish time over all jobs (Makespan) ;

! Keywords: @OLE, Changeovers, Flowshop, LINGO, Makespan, Scheduling, Sequencing, Setups;
SETS:
    JOB;
    MACHINE;
    MxJ( MACHINE, JOB): DURN, STIME, ETIME;
    MxJxJ( MACHINE, JOB, JOB): ChngTm, ZPred;
ENDSETS
DATA:
! Get input data internally;
!Cint;  UseExcel = 0; ! Using Excel ? ;
! Get names of the jobs;
!Cint;  JOB = A1 A2 A3 B1 B2 B3;
! Get names of the machines;
!Cint;  MACHINE = Blend Tablet;
! Duration of each job (col) on each machine( row);
!Cint;  DURN =
    2 2 2 2 2 2
    3 3 3 3 3 3
;
! Changeover times, machine, predecessor job, successor job;
!Cint;  ChngTm =
    1 1 1 2 2 2
    1 1 1 2 2 2
    1 1 1 2 2 2
    1 1 1 2 2 2
    1 1 1 2 2 2
    1 1 1 2 2 2
    1 1 1 2 2 2
    1 1 1 2 2 2
    1 1 1 1 1 1
    1 1 1 1 1 1
    1 1 1 1 1 1
;
!Cint;  ULSTIME = 99; ! Upper limit on difference in two start times,
! for speed, should be small as valid;

! Get input data from Excel ( Should be just one file open in Excel).
! Range names in Excel should match variable names in LINGO;
!CXcel  UseExcel = 1; ! Using Excel ? ;
!CXcel  JOB = @OLE(); ! Get names of the jobs;
!CXcel  MACHINE = @OLE(); ! Get names of the machines;
!CXcel  DURN = @OLE(); ! Get duration of the processing times on each machine;
!CXcel  ChngTm = @OLE(); ! Get changeover times;
!CXcel  ULSTIME = @OLE(); ! Get upper limit on time between any two start times,
! for speed, should be small as valid;
ENDDATA

SUBMODEL FloSched:
! Zpred( m, i, j) = 1 if job i precedes j on machine m,
! Zpred( m, i, i) = 1 if i is first job on machine m,
! STime( m, j) Start time on machine m of job j;
! NMACH = @SIZE( MACHINE); ! Get number of machines;
! Minimize the maximum finish time;
MIN = FinMax;
@FOR( JOB( j):
    FinMax >= Etime( NMACH, j);
);
! Job j cannot start earlier on machine m than its finish time on m-1;
@FOR( MxJ( m, j) | m #GT# 1:
    STIME( m, j) >= STIME( m-1, j) + DURN( m-1, j);
);

```

```

@FOR( MxJ( m, j):
! Job j must be preceded by some job i on machine m;
@SUM( JOB( i): ZPred( m, i, j)) = 1;
! Job j has at most one successor on machine m;
@SUM( JOB( i) | i #NE# j: ZPred( m, j, i)) <= 1;

! Earliest start of j on machine m is >= start of predecessor job i,
+ its duration + changeover time from i to j;
@FOR( JOB( i) | i #NE# j:
    STime( m, j) >= STime( m, i) + ( Durn( m, i) + Chngtm( m, i, j) ) * ZPred( m, i, j)
    - ULSTime * ( 1 - Zpred( m, i, j ))
);
ETIME( m, j) = STIME( m, j) + DURN( m, j); ! Compute end time on each machine, each
job;
);

! There is exactly one first job on each machine;
@FOR( MACHINE( m):
    @SUM( JOB( j): ZPred( m, j, j)) = 1;
);

! The ZPRED must be binary ( 0 or 1);
@FOR( MxJxJ( m, i, j): @BIN( ZPred( m, i, j)));
ENDSUBMODEL

CALC:
@SET( 'TERSEO', 2); ! Output level (0:verbose, 1:terse, 2:only errors, 3:none);
@SET( 'IPTOLR', 0.1); ! Set IP ending relative optimality tolerance(Should be >0);
@SET( 'TIM2RL', 30); ! Time in seconds to apply IPTOLR tolerance;

@SOLVE( FloSCHED); ! Solve a specified submodel;

@WRITE( ' Machine      Job      Start_time  End_time', @NEWLINE( 1));
@FOR( MXJ( m, j):
    @WRITE( @FORMAT( Machine( m), '9s'), ' ', @FORMAT( Job( j), '9s'), ' ',
            @FORMAT( STIME( m, j), '9.0f') , ' ', @FORMAT( ETIME( m, j), '9.0f'), @NEWLINE(
1));
    );

@IFC( UseExcel:
! Send the Start times of each job on each machine back to the spreadsheet;
@OLE() = STime;
! Send the End times of each job on each machine back to the spreadsheet;
@OLE() = ETime;
);

ENDCALC

```