APPENDIX

Tecnomatix coding for the model:

- The setting of parts that has been assembled

waituntilAssembly.Occupiedprio 1;@.move

- The summary list of station time

TaskTime_List[2,1]:=Station1_EF1.ProcTime+2;TaskTime_List[2,2]:=Station1_LH1.ProcTime+2;……TaskTime_List[3,26]:=Station10_LH2.ProcTime+2;TaskTime_List[3,27]:=Station10_EB2.ProcTime+2;

- Tasks placement

TableFile1[2,1]:=Station1_EF1.ProcTime+2;TableFile1[2,2]:=Station1_EF2.ProcTime+2;……TableFile27[2,1]:=Station10_EB1.ProcTime+2;TableFile27[2,2]:=Station10_EB2.ProcTime+2;

- The overall proportion of units produced (qm)

qm_Model_A:=(Model_A_Innova/TotalDemand);

qm_Model_B:=(Model_B_Fortuner/TotalDemand);

-Calculation of task time total

TOTAL_TASK_TIME_INNOVA:=TableFile1[2,1]+……..+TableFile27[2,1]

TOTAL_TASK_TIME_FORTUNER:=TableFile1[2,2]+……+TableFile27[2,2]

- Waiting time

WaitingTime_List[2,1]:=Cycle_Time-(Station1_EF1.ProcTime+2);…….WaitingTime_List[2,27]:=Cycle_Time-(Station10_EB2.ProcTime+2);WaitingTime_List[3,1]:=Cycle_Time-

(Station1_EF1.ProcTime+2);…….WaitingTime_List[3,27]:=Cycle_Time-(Station10_EB2.ProcTime+2);
WaitingTime_List[4,1]:=WaitingTime_List[2,1]+…….+WaitingTime_List[2,27];
WaitingTime_List[5,1]:=WaitingTime_List[3,1]+…….+WaitingTime_List[3,27];

- Weighted line efficiency
Weighted_Line_Efficiency_Percentages:=((qm_Model_A*(TOTAL_TASK_TIME_INNOVA)/(Cycle_Time*NumberofWorkstations))+(qm_Model_B*(TOTAL_TASK_TIME_FORTUNER)/(Cycle_Time*NumberofWorkstations)))*100;

- Weighted smoothness index
Weighted_Smoothness_index:=sqrt((qm_Model_A*((TaskTimeList[2,1]-TaskTime_List[3,9])*(TaskTime_List[2,1]-TaskTime_List[3,9])+……+(TaskTime_List[2,27]-TaskTime_List[3,9])*(TaskTime_List[2,27]-TaskTime_List[3,9])))/27)+sqrt((qm_Model_B*((TaskTime_List[3,1]-TaskTime_List[3,9])*(TaskTime_List[3,1]-TaskTime_List[3,9])+…….+(TaskTime_List[3,27]-TaskTime_List[3,9])*(TaskTimeList[3,27]-TaskTime_List[3,9])))/27);

- Cycle time constraint
CT_Max:=Planning_Horizon/TotalDemand;
CT_Min:=(qm_Model_A*(TOTAL_TASK_TIME_INNOVA))/27+(qm_Model_B*(TOTAL_TASK_TIME_FORTUNER))/27;

Optimization program of MTALB type II problem:

is
i,k,m:integer;
station:object;
startX,startY:integer;
p:object;
contCells:integer;

```
tab:table[string,time];
remProcTime:time;
row:integer;
preds:list;
assigned:boolean;
do
startX:=240;
startY:=240;
p:=source;
tab.create;
partData.delete({3,1}..{3,*});
--clean the old version from source, drain, station
for i:= self.~.numNodesdownto 1 loop
        ifself.~.node(i).class.name="Drain" or
        self.~.node(i).class.name="Station" then
                self.~.node(i).deleteObject;
        end;
next;
--create all stations from partData from Col 4++
for i:=4 to partData.xDim loop
        --create a new station
station:=.ApplicationObjects.UserObjects.Station.createObject(self.~,startX,starty);
        station.name:=partData[i,0];
        startX:=startX+60;
        .MaterialFlow.Connector.connect(p,station);
        p:=station;
        --look for not assigned parts
        --prio 1. MUST assembled here --field Station,part is occupied and number of
filled cells right until xDim is zero
        for k:=1 to partData.ydim loop
                --not assigned
                ifpartData[3,k] = void and partData[Station.name,k] /= void then
                        --marked
```

```
                    --look at all other cols in this row
                    contCells:=0;
                    for m:= partData.getColumnNo(Station.name)+1 to
partData.xdim loop
                        ifpartData[m,k] /= void then
                            contCells:=contCells+1;
                        end;
                    next;
                    ifcontCells=0 then
                        --must be assembled here !!
                        partData[3,k]:=station.name;
station.workContent.writeRow(1,station.workContent.yDim+1,partData[1,k],partData[2
,k]);
                    end;
                end;
            next;
            --prio 2. Search for the part with the lowest number of alternatives and max
possible assembly time
            --write all possible parts into a table [part, time, num alternatives]
            tab.delete;
            remProcTime:=root.cycleTime- station.workcontent.sum({2,1}..{2,*});
            for k:=1 to partData.ydim loop
                --not assigned
                if station.name="Station3" then
                --      debug;
                end;
                ifpartData[3,k] = void and partData[Station.name,k] /= void then
                    --write all possible operations into a table
                    ifpartData[2,k] <= remProcTime then
                        --check the position of the part in the assembly graph
                        --all predessessors need to be assembled
            preds:=assemblyOrder.getPartPredecessors(partData[1,k]);
                        --if the ist is empty, no predessor
```

```
                                ifpreds.dim=0 then
tab.writeRow(1,tab.yDim+1,partData[1,k],partData[2,k]);
                                else
                                        --assign only parts, when all predessessors are
assigned
                                        assigned:=true;
                                        for m:=1 to preds.dim loop
                                                ifpartData[3,preds.read(m)] = void then
                                                        assigned:=false;
                                                end;
                                        next;
                                        if assigned=true then
tab.writeRow(1,tab.yDim+1,partData[1,k],partData[2,k]);
                                        end;
                                end;
                        end;
                end;
        next;
        --sort tab second column "down"
        --tab.sort(2,"down"); sort mixes the assembly order
        --fill the station with content
        for m:=1 to tab.yDim loop
                --check remaining cycle time
                remProcTime:=root.cycleTime- station.workcontent.sum({2,1}..{2,*});
                        ifremProcTime>=tab[2,m] then
                        --assign to station
                        partData.setCursor(1,1);
                        partData.find({1,1}..{1,*},tab[1,m]);
                        partData[3,partData.cursorY]:=station.name;
                        --write part in work content
station.workContent.writeRow(1,station.workContent.yDim+1,tab[1,m],tab[2,m]);
                end;
        next;
```

```
        --set the sum of assembly times as processTime of the station
        station.procTime:=station.workcontent.sum({2,1}..{2,*});
next;
--connect the last station with a drain
p:=.MaterialFlow.Drain.createObject(self.~,startX,starty);
.MaterialFlow.Connector.connect(station,p);end;
```