CHEMICAL ENGINEERING TRANSACTIONS
VOL. 32, 2013
Chief Editors: Sauro Pierucci, Jiří J. Klemeš

ISBN 978-88-95608-23-5; ISSN 1974-9791

A publication of
AIDIC
The Italian Association of Chemical Engineering
Online at: www.aidic.it/cet

DOI: 10.3303/CET1332231

General Framework for Automated Manufacturing Systems: Multiple Hoists Scheduling Solution

Adrián M. Aguirre[a], Carlos A. Méndez[a*], Álvaro García-Sánchez[b], Miguel Ortega-Mier[b], Pedro M. Castro[c]

[a]INTEC (UNL - CONICET), Güemes 3450, 3000 Santa Fe, Argentina
[b]ETSI Industriales, (UPM), C/José Gutierrez Abascal 2, 28006 Madrid, Spain
[c]UMOSE (LNEG), 1649-038 Lisboa, Portugal
*cmendez@intec.unl.edu.ar

This work presents a generic MILP-based framework for the scheduling of Hoist transportation devices in a complex Automated Manufacturing System. The main contribution of this approach relies on the possibility to tackle multiple hoists in a multiproduct multistage batch process under different production schemes considering: heterogeneous production recipes, multiple units per production stage, possible recycle flows, sequence-dependent transfers and flexible processing times. The effectiveness of this approach was tested in a real industrial application example which was solved with acceptable computational effort.
Keywords: MILP-based approach, Automated Manufacturing Systems, Open-shop Scheduling problems, Sequence-dependent transferring times, Real-world industrial application.

1. Introduction

Hoist Scheduling Problems (HSP) represent one of the most studied problems for practitioners and researches in Automated Manufacturing Systems. Many recent works in this area are focusing on providing efficient solution methods based on MILP models considering only a single Hoist in a single track, by trying to minimize the total time of the production schedule (e.g., Zhao et al., 2013). Other developments, that might consider re-entrant flows (Che et al., 2012) and parallel machines (Fröhlich, Steneberg, 2011), were oriented to solve the problem of multiple hoists in the same track, avoiding possible collisions between those transportation devices. For more information about HSP problems and features see also Crama (1997).

The problem presented here, tries to emphasize the issue of two-hoists located in the same track by considering the use of different adjacent zones per each device. According to this, hoists must be allocated in different zones *z* *(z=z1,...,R)* that comprise several units *j* *(j=j1,...,M)*. It is worth to remark that, this kind of division can be applied only in systems in which units can be clearly separated in non-overlapping zones (Liu, Jiang, 2005) but for a common unit or stage for every two zones (see Fig. 1).

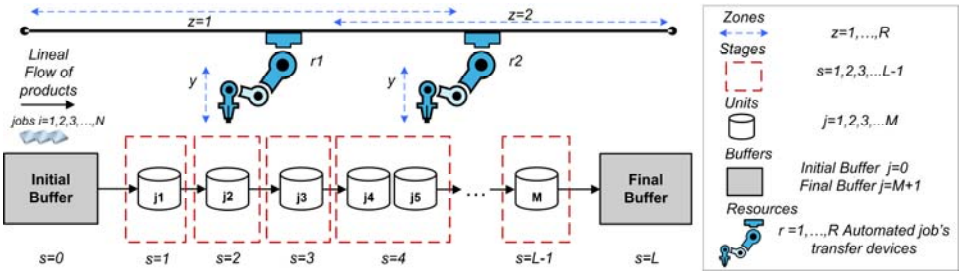

Fig. 1: Lineal configuration of Automated Systems with multiple Hoists in a single track divided by zones.

Please cite this article as: Aguirre A.M., Mendez C.A., Garcia Sánchez Á., Ortega-Mier M., Castro P., 2013, General framework for automated manufacturing systems: multiple hoists scheduling solution, Chemical Engineering Transactions, 32, 1381-1386 DOI: 10.3303/CET1332231

The general mathematical approach developed for this problem can handle, a) different production recipes $Seq_{(i)}$, b) alternative units *j* per production stage *s*, c) multiple resources *r (r=r1,...,T)*, also called robots, in the same single track divided by zones, d) load transferring times (π^{load}), e) sequence-dependent free transfer times (π^{free}), f) possible re-entrant or recycle flows to the same unit g) flexible processing times $(t_{(i,s)})$ and h) stringent "Zero Wait" (ZW) and "Non-Intermediate Storage" (NIS) policies (see Fig. 2).

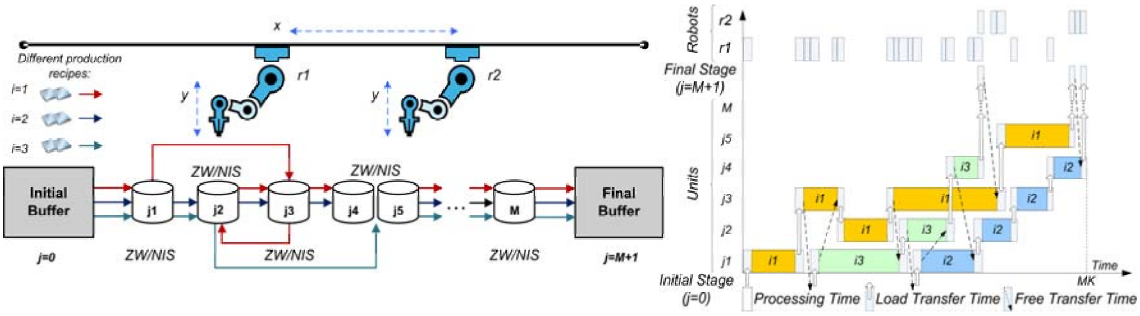

Fig. 2: Example of Automated Systems under different production recipes with recycles and parallel units.

The main aim is to find the best production and transportation schedule that minimize the total completion time of all jobs *i (i=i1,...,N)* in the system, widely known as *Makespan* criterion *(MK)*. For this, unit and resource assignment variables $w_{(i,s,j)}$, $q_{(i,s,r)}$ and sequencing binary variables $X_{(i,i',s,s')}$,$Y_{(i,i',s,s')}$ are introduced in the model in order to correctly synchronize job's and transfer's operations.

2. General MILP-based Framework

The solution presented is derived from a General-Precedence MILP-based model of Aguirre et al. (2011) developed for a particular flow-shop scheduling problem in the semiconductor industry. Then, based on a previous work (Kopanos et al., 2010) proposed for a particular multi-product multi-stage batch pharmaceutical process, the previous model was extended to handle job-shop scheduling issues in the aircraft-part manufacturing system (Aguirre et al., 2012). Additional information about MILP-based methods for industrial batch process scheduling can be found in Hegyháti, Friedler (2010).
In this work we develop a more tightened model in terms of binary variables to arise open-shop (flow-shop and job-shop) problems with transferring tasks in multiple Hoist or Robots.The principal contribution of this novel general approach relies in the possibility to consider different production schemes, such as, multiple resources, parallel units, re-entrant flows and sequence-dependent transferring tasks, without generating unnecessary binary variables.

2.1 Model Assumptions

The following model is able to handle different robots in the system with the following assumptions:

1. Robots are assigned to different zones in the same track.
2. Consecutive zones can only overlap in an adjacent unit and no collisions occur in this unit during transferring tasks.
3. Pick-up and drop-down movements are included into the load transferring time.
4. Immediately after a load transfer task is done the robot has to begin the next free move to the following unit in their load transfer sequence.
5. Robots are located in the opposite sites of the line at the beginning and at the end of the production time.
6. Robots can only perform one transfer task at a time.
7. Some stages can have alternative production units (parallel units).
8. All units have a single capacity.

2.2 Notation

I, S, J, R	Set of jobs *(i=i$_1$,...,N)*, stages *(s=s$_0$,...,L)*, unit *(j=j$_0$,...,M)*, robots *(r=r1,...,T)*,
S_i	Subset of stages of job *i*,
L_i	Last stage in the processing sequence of job *i*,
$J_{i,s}$	Subset of units that can be used to perform task *i,s,*
$R_{i,s}$	Subset of tasks *i,s* transferred by robot *r*,
H_r	Initial position of robot *r* in the production line,

$Seq_{(i)}$	Processing sequence or production recipe of job *i*,
$Ts_{(i,s)}$, $Tf_{(i,s)}$	Start time and Completion time of task *i,s*,
$t_{(i,s)}$	Processing time of task *i,s*,
$t^{min}{}_{(i,s)}$, $t^{max}{}_{(i,s)}$	Minimum and Maximum processing time of task *i,s*,
$\pi^{load}{}_{(i,s)}$, $\pi^{free}{}_{(i,s)}$	Loaded and Free Transferring time from of task *i,s*,
$\pi^{min}{}_{(i,s)}$, $\pi^{max}{}_{(i,s)}$	Minimum and Maximum loaded transfer time of task *i,s*,
$\pi^{abs}{}_{(j,j')}$	Absolute distance between units *j* and *j'*,
$\pi^{seq\text{-}dep}{}_{(i,i',s,s')}$	Sequence-dependent transfer times from task *i',s'* to task *i,s*,
$X_{(i,i',s,s')}$, $Y_{(i,i',s,s')}$	Job sequencing and Transfer sequencing binary variables,
$K_{(i,i',s,s',r)}$	Sequencing variables for immediate-precedence transfers,
$w_{(i,s,j)}$, $q_{(i,s,r)}$	Assignment binary variables of task *i,s* to machine *j* and/or to robot *r*,
$Pos_{(i,s,r)}$	Position of task *i,s* in the transferring sequence on a single resource *r*,
M_T, MK	Large number (Big-M parameter) and Makespan Criterion,

2.3 Objective Function: "*Makespan Minimization*"

Equation (1) is derived to determine the maximum completion time of all jobs in the system.

$$MK \geq Ts_{(i,s)} \qquad \forall i \in I, s \in S_i : (s = L_i) \tag{1}$$

2.4 Assignment constraints:

This model considers different units *j* and also different robots *r* to perform all transferring tasks *i,s* in different subsets $J_{i,s}$ and $R_{i,s}$. According to this, the model has to decide if task *i,s* is done in unit *j* by adopting $w_{(i,s,j)}=1$ and also if task *i,s* is transferred by robot *r* by adopting $q_{(i,s,r)}=1$. In order to do these, equations (2-3) are proposed to define that only one unit *j* and only one robot *r*, of the possible ones in $J_{i,s}$ and $R_{i,s}$ respectively, can be used to process and to move task *i,s* in the system.

$$\sum_{j \in J_{i,s}}^{M+1} w_{(i,s,j)} = 1 \qquad \forall i \in I, s \in S_i \tag{2}$$

$$\sum_{r \in R_{i,s}}^{T} q_{(i,s,r)} = 1 \qquad \forall i \in I, s \in S_i \tag{3}$$

2.5 Flexible timing constraints:

Flexible processing times are considered along the process in each production stage under ZW policy by Eq.(4). Thus, the processing of task *i,s* in a particular unit *j* has to be produced between a minimum a maximum time, as is expressed in Eq.(5). After the processing time is reached, job *i* has to be immediately transferred to the following stage without extra holding time.

$$Tf_{(i,s)} = Ts_{(i,s)} + t_{(i,s)} \qquad \forall i \in I, s \in S_i : s \neq L_i, ZW \tag{4}$$

$$t^{\min}{}_{(i,s)} \leq t_{(i,s)} \leq t^{\max}{}_{(i,s)} \qquad \forall i \in I, s \in S_i \tag{5}$$

2.6 Transfer constraints:

Flexible transferring times are also considered in the entire process. Due to no intermediate buffers exists among the stages, Non-Intermediate Storages (NIS) policy must be followed in the system. These features of the process are expressed by the following Eqs. (6-8).

$$Ts_{(i,s)} = Tf_{(i,s-1)} + \pi^{load}{}_{(i,s)} \qquad \forall i \in I, s \in S_i : s > 1, NIS \tag{6}$$

$$Ts_{(i,s)} \geq \pi^{load}{}_{(i,s)} \qquad \forall i \in I, s \in S_i : s = 1 \tag{7}$$

$$\pi^{\min}{}_{(i,s)} \leq \pi^{load}{}_{(i,s)} \leq \pi^{\max}{}_{(i,s)} \qquad \forall i \in I, s \in S_i \tag{8}$$

2.7 Job sequencing constraints in a single unit:

Sequencing decisions between different tasks *i,s* and *i',s'* in a single unit *j* along the process are made by binary variable $X_{(i,i',s,s')}$ in Eqs. (9-10).

$$X_{(i,i',s,s')} = \begin{cases} 1 & \text{if task } i,s \text{ is processed after task } i',s' \text{ in the same unit } j \\ 0 & \text{otherwise} \end{cases}$$

$$Ts_{(i,s)} \geq Tf_{(i',s')} + \pi^{load}{}_{(i,s)} + \pi^{load}{}_{(i',s'+1)} - M_T(1 - X_{(i,i',s,s')}) - M_T(2 - w_{(i,s,j)} - w_{(i',s',j)})$$
$$\forall i, i' \in I : (i > i'), s \in S_i, s' \in S_{i'}, j \in J_{i,s}, j \in J_{i',s'} \tag{9}$$

$$Ts_{(i',s')} \geq Tf_{(i,s)} + \pi^{load}{}_{(i',s')} + \pi^{load}{}_{(i,s+1)} - M_T(X_{(i,i',s,s')}) - M_T(2 - w_{(i,s,j)} - w_{(i',s',j)})$$
$$\forall i, i' \in I : (i > i'), s \in S_i, s' \in S_{i'}, j \in J_{i,s}, j \in J_{i',s'} \tag{10}$$

2.8 Transfer sequencing constraints:

Sequencing transfer decisions related to a pair of transfer tasks *i,s* and *i',s'* performed in different units are modeled by binary variable $Y_{(i,i',s,s')}$ in Eqs. (11-12).

$$Y_{(i,i',s,s')} = \begin{cases} 1 & \text{if transfer } i,s \text{ is processed after transfer } i',s' \text{ in the in the same robot } r \\ 0 & \text{otherwise} \end{cases}$$

$$Ts_{(i,s)} \geq Ts_{(i',s')} + \pi^{load}{}_{(i,s)} + \pi^{free}{}_{(i,s)} - M_T(1 - Y_{(i,i',s,s')}) - M_T(2 - q_{(i,s,r)} - q_{(i',s',r)})$$
$$\forall i,i' \in I : (i > i'), s \in S_i, s' \in S_{i'}, r \in R_{i,s}, r \in R_{i',s'} \quad (11)$$

$$Ts_{(i',s')} \geq Ts_{(i,s)} + \pi^{load}{}_{(i',s')} + \pi^{free}{}_{(i',s')} - M_T(Y_{(i,i',s,s')}) - M_T(2 - q_{(i,s,r)} - q_{(i',s',r)})$$
$$\forall i,i' \in I : (i > i'), s \in S_i, s' \in S_{i'}, r \in R_{i,s}, r \in R_{i',s'} \quad (12)$$

2.9 Immediate-precedence constraints using general-precedence information:

Position variable ($Pos_{(i,s,r)}$) is defined by Eqs. (13-16) as the absolute location of each transfer task *i,s* in each robot *r* sequence. It is calculated by using the information of the general precedence binary variable $Y_{(i,i',s,s')}$ in Eqs. (13-14) and also by the information of assignment binary variable $q_{(i,s,r)}$ in Eqs. (15-16).

$$Pos_{(i,s,r)} \geq Pos_{(i',s',r)} + 1 - M_T(1 - Y_{(i,i',s,s')}) - M_T(2 - q_{(i,s,r)} - q_{(i',s',r)})$$
$$\forall i,i' \in I : (i \geq i'), s \in S_i, s' \in S_{i'} : (i,s) \neq (i',s'), r \in R_{i,s}, r \in R_{i',s'} \quad (13)$$

$$Pos_{(i',s',r)} \geq Pos_{(i,s,r)} + 1 - M_T(Y_{(i,i',s,s')}) - M_T(2 - q_{(i,s,r)} - q_{(i',s',r)})$$
$$\forall i,i' \in I : (i \geq i'), s \in S_i, s' \in S_{i'} : (i,s) \neq (i',s'), r \in R_{i,s}, r \in R_{i',s'} \quad (14)$$

$$Pos_{(i,s,r)} \leq \sum_{i' \in I}^{N} \sum_{s' \in S_{i'}}^{L_i} \sum_{r \in R_{i',s'}}^{T} q_{(i',s',r)} \qquad \forall i \in I, s \in S_i, r \in R_{i,s} \quad (15)$$

$$q_{(i,s,r)} \leq Pos_{(i,s,r)} \leq M_T q_{(i,s,r)} \qquad \forall i \in I, s \in S_i, r \in R_{i,s} \quad (16)$$

Then, an additional continuous variable $K_{(i,i',s,s')}$ is proposed by Eqs. (17-18) to determine the immediate-precedence of a transfer task *i,s* in the robot sequence.

$$K_{(i,i',s,s',r)} = 0 \quad \text{if transfer } i,s \text{ is inmediately processed after transfer } i',s' \text{ in robot } r$$

$$K_{(i,i',s,s',r)} = Pos_{(i,s,r)} - Pos_{(i',s',r)} - 1 + M_T(1 - Y_{(i,i',s,s')}) + M_T(2 - q_{(i,s,r)} - q_{(i',s',r)})$$
$$\forall i,i' \in I : (i \geq i'), s \in S_i, s' \in S_{i'} : (i,s) \neq (i',s'), r \in R_{i,s}, r \in R_{i',s'} \quad (17)$$

$$K_{(i',i,s',s,r)} = Pos_{(i',s',r)} - Pos_{(i,s,r)} - 1 + M_T(Y_{(i,i',s,s')}) + M_T(2 - q_{(i,s,r)} - q_{(i',s',r)})$$
$$\forall i,i' \in I : (i \geq i'), s \in S_i, s' \in S_{i'} : (i,s) \neq (i',s'), r \in R_{i,s}, r \in R_{i',s'} \quad (18)$$

2.10 Sequence-dependent and free transferring times:

The information presented below and the values of variable $K_{(i,i',s,s',r)}$ will be used to determine the sequence-depending free transferring times by Eq. (19). Also, sequence dependent transferring times $\pi^{seq\text{-}dep}{}_{(i,i',s,s')}$ is determined by the model by Eq.(20) according to the absolute distance between the initial and the departure units *j'* and *j* of transferring tasks *i',s'* and *i,s-1* defined in parameter $\pi^{abs}{}_{(j,j')}$. In the same way, Eq. (21) determines the first free move of each robot from their initial position $H_r=j'$ to unit j always if transfer task *i,s* is done first $Pos_{(i,s,r)}=1$, if unit *j* is the previous unit in the recipe $w_{(i,s-1,j)}=1$ and if this transfer *i,s* has done in robot *r* according to $q_{(i,s,r)}=1$.

$$\pi^{free}{}_{(i,s)} \geq \pi^{seq-dep}{}_{(i,i',s,s')} - M_T(K_{(i,i',s,s',r)}) - M_T(2 - q_{(i,s,r)} - q_{(i',s',r)})$$
$$\forall i,i' \in I, s \in S_i, s' \in S_{i'} : (i,s) \neq (i',s'), r \in R_{i,s}, r \in R_{i',s'} \quad (19)$$

$$\pi^{seq-dep}{}_{(i,i',s,s')} \geq \pi^{abs}{}_{(j,j')} - M_T(2 - w_{(i,s-1,j)} - w_{(i',s',j')})$$
$$\forall i,i' \in I, s \in S_i, s-1 \in S_i, s' \in S_{i'} : (i,s) \neq (i',s'), j \in J_{i,s-1}, j' \in J_{i',s'} \quad (20)$$

$$\pi^{free}{}_{(i,s)} \geq \pi^{abs}{}_{(j,j')} - M_T(Pos_{(i,s,r)} - 1) - M_T(2 - q_{(i,s,r)} - w_{(i,s-1,j)})$$
$$\forall i \in I, s \in S_i, s-1 \in S_i, j \in J_{i,s-1}, j' = H_r, r \in R_{i,s} \quad (21)$$

Motivating example

2.11 Problem data and Results

An industrial application example of real-life operations in the surface-treatment process of metal components in an aircraft industry is presented here. Related MILP-based works, such as Zhou, Li, (2009), were proposed first to solve similar problems in the manufacturing of Printed Circuit Board (PCB) in electroplating plants trying to minimize the total cycle time of the entire system.

In surface-treatment processes, jobs i_1-i_6 have to be scheduled in units j_0-j_{37} following their own production recipes $Seq_{(i)}$. In this case three possible recipes $Seq_{(i)}=[1,2,3]$ are allowed for each job and these are $Seq=\{i1,i2\}=1$, $Seq=\{i3,i4\}=2$ and $Seq=\{i5,i6\}=3$. The information of processing times $t_{(i,s)}$ of every task *i,s* and the sequence of units *j* followed by each recipe $Seq_{(i)}$ are presented in *Table 1*. This table shows that all jobs have to start the process in unit j_0 and finish it in unit j_{37}. In addition, recycle flows can occur in unit j_5 while j_{16} - j_{17} and j_{25} - j_{26} can be used as parallel units.

Table 1. Minimum $t^{min}_{(i,s)}$ and Maximum $t^{max}_{(i,s)}$ processing times of task i,s in unit j

Seq	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
1	j_0:0'	j_3:10'-15'	j_5:5'-15'	j_4:1'	j_5:10'-15'	j_7:10'	j_{25-26}:30-60'	j_{35}:15'-60'	j_{37}:0'
2	j_0:0'	j_3:10'-15'	j_5:5'-10'	j_7:1'-5'	j_9:5'-10'	j_{16-17}:30-60'	j_{35}:15'-60'	j_{37}:0'	
3	j_0:0'	j_3:10'-15'	j_5:5'-10'	j_7:1'-5'	j_{35}:15'-60'	j_{37}:0'	-	-	

Flexible transfer times between $\pi^{min}_{(i,s)}$ and $\pi^{max}_{(i,s)}$ are considered for every transfer (see *Table 2*). While, sequence-dependent transferring times are estimated by Eq.(19) according to the absolute distance between units of consecutive transfer tasks *i',s'* and *i,s*. Thus, a positive variable $\pi^{seq\text{-}dep}_{(i,i',s,s')}$ could be estimated by $\pi^{abs}_{(j,j')} = abs(j\text{-}j')*0.05[min.]$, always if task *i,s-1* if performed in unit *j* and task *i',s'* in unit *j'* by considering $w_{(i,s\text{-}1,j)}=1$ and $w_{(i',s',j')}=1$.

Table 2. Minimum $\pi^{min}_{(i,s)}$ and Maximum $\pi^{max}_{(i,s)}$ transferring times of task i,s from the previous stage i,s-1

Seq	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
1	π:0'	π:1-6'	π:1-6'	π:1-6'	π:1-6'	π:1-6'	π:3-6'	π:2-6'	π:1-6'
2	π:0'	π:1-6'	π:1-6'	π:1-6'	π:1-6'	π:2-6'	π:3-6'	π:1-6'	-
3	π:0'	π:1-6'	π:1-6'	π:1-6'	π:3-6'	π:1-6'	-	-	-

The main statistics of this problem considering three different instances: without robots, with a single and two robots in zones, are presented in *Table 3*.

Table 3. Statistics and Results of the problem analyzed

Units x Jobs	Statistics	Reduced MILP model (without robots)	Full-space MILP model (single robot)	Full-space MILP model (two robots)
	Binary Var.	140	670	340
	Cont. Var.	137	3895	3011
35x6	Equations	436	13452	9096
	MK [min.]	157	161.2	160.05
	CPU [sec.]	0.5	1650	21

Solutions reported by using Gurobi 3.0 in a PC Intel Core 2 Quad 2,5 GHz with parallel processing in 4 threads.

The results in *Table 3*, illustrates that this model is able to solve the problem without any robot and also with one and multiple robots finding optimal results in reasonable computational time. Also it is important to remark that the full-space model considering only a single robot requires much more CPU time to find optimal solutions due to the number of transfer decisions that have to be made along the line. In other hand, when the line is divided in zones, the number of binary variables decrease considerably an also the CPU time required to solve the problem. A detailed schedule of the problem considering two robots in adjacent zones is reported in Figure 3.

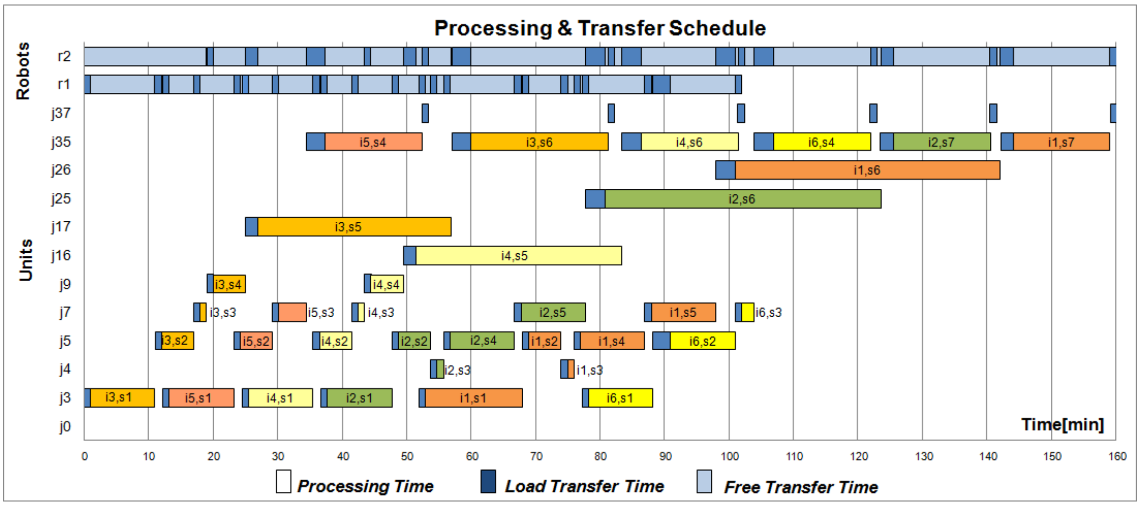

Fig. 3: Solution Schedule considering two hoists (r1 and r2) in adjacent zones z1:j0-j7 and z2:j7-j37.

3. Conclusions

An MILP-based procedure was developed for the scheduling of multiple products in multi-stage automated manufacturing systems. The solution approach allows finding optimal schedules of processing and transfer operations taking into account sequence-dependent transferring times, parallel units in some stages, re-entrant flows to the same unit and also considering multiple hoists or robots in a “zoned” single track line avoiding possible collisions. Finally, our exact mathematical approach was tested in a real-world example, providing an optimal result with modest CPU effort.

Acknowledgments

Financial support received from AECID under Grant PCI-A1/044876/11, from CONICET under Grant PIP-2221, from UNL under Grant PI-66-337 and from the Erasmus Mundus ARCOIRIS Scholarship program is fully appreciated.

References

Aguirre, A. M., Méndez, C. A., Castro, P. M., 2011. A novel optimization method to automated wet-etch station scheduling in Semiconductor Manufacturing Systems, Computer Chemical Engineering, 35, 2960-2972.

Aguirre, A. M., Méndez, C. A., Castro, P. M., De Prada, C., 2012. MILP-based approach for the scheduling of Automated Manufacturing System with sequence-dependent transferring times. Computer-Aided Chemical Engineering, 30, 477-481.

Che, A., Chabrol, M., Gourgand, M., Wang, Y., 2012. Scheduling multiple robots in a no-wait re-entrant robotic flowshop. International Journal of Production Economics, 135, 199-208.

Crama, Y., 1997. Combinatorial optimization models for production scheduling in automated manufacturing systems. European Journal of Operational Research 99 (1), 136-153.

Hegyháti, M., Friedler, F., 2010. Overview of Industrial Batch Process Scheduling. Chemical Engineering Transactions, 21, 895-899.

Fröhlich, R., Steneberg, S., 2011. Transient Multiple Hoist Scheduling of Consecutive Cyclic Schedules. Proceeding of the IEEE International Conference on Automation Science and Engineering, 357-362.

Kopanos, G.M., Méndez, C.A., Puigjaner, L., 2010. Solving Scheduling Problems in a Multi-stage Multi-product Batch Pharmaceutical Industry. Chemical Engineering Transaction, 21, 511-516.

Liu, J., Jiang, Y., 2005. An Efficient Optimal Solution to the Two-Hoist No-Wait Cyclic Scheduling Problem. Operations Research 53(2), 313-327.

Zhao, C., Fu, J., Xu, Q., 2013. Production-ratio oriented optimization for multi-recipe material handling via simultaneous hoist scheduling and production line arrangement. Computer and Chemical Engineering, 50, 28-38.

Zhou, Z., Li, L., 2009. A solution for cyclic scheduling of multi-hoists without overlapping. Annals of Operations Research, 168, 5–21