

## SCHEDULING COPPER REFINING AND CASTING OPERATIONS BY MEANS OF HEURISTICS FOR THE FLEXIBLE FLOW SHOP PROBLEM

Lorena Pradenas<sup>1\*</sup>, Abel Campos<sup>2</sup>, Jesús Saldaña<sup>3</sup> and Víctor Parada<sup>4</sup>

Received October 7, 2008 / Accepted April 18, 2011

**ABSTRACT.** Management of the operations in a copper smelter is fundamental for optimizing the use of the plant's installed capacity. In the refining and casting stage, the operations are particularly complex due to the metallurgical characteristics of the process. This paper tackles the problem of automatic scheduling of operations in the refining and casting stage of a copper concentrate smelter. The problem is transformed into a flexible flow shop problem and to solve it, an iterative method is proposed that operates in two stages: in the first stage, a sequence of jobs is constructed that configures the lots, and in the second, the constructed solution is improved by means of simulated annealing. Fifteen test problems are used to show that the proposed algorithm improves the makespan by an average of 9.42% and the mean flow time by 12.19% with respect to an existing constructive heuristic.

**Keywords:** refining and casting, flexible flow shop, simulated annealing.

### 1 INTRODUCTION

In a copper concentrate smelter, a flow of wet *sulfide ore* is received, and through a metallurgical process, copper is produced. The process involves various typical operations, such as the storage and preparation of the load, melting, conversion, refining and casting, removing slag, and absorption of gases. In the first operation, the wet ore is dried in a rotary furnace, and in smelting, the dry ore is fed into reactors where it is subjected to high temperatures to purify it. As a result, blister copper is obtained that contains 62 to 75% copper, which is the raw material for the conversion operation in which a new purification process gives rise to a product that contains about

---

\*Corresponding author

<sup>1</sup>Industrial Engineering Department, University of Concepcion, Casilla 160-C, Correo 3, Concepción, Chile.  
E-mail: lpradena@udec.cl

<sup>2</sup>Industrial Engineering Department, University of Concepcion, Casilla 160-C, Correo 3, Concepción, Chile.  
E-mail: abcampos@udec.cl

<sup>3</sup>Industrial Engineering Department, University of Concepcion, Casilla 160-C, Correo 3, Concepción, Chile.  
E-mail: jessaldana@udec.cl

<sup>4</sup>Informatic Engineering Department, University of Santiago of Chile, 3659 Av. Ecuador, Santiago, Chile.  
E-mail: victor.parada@usach.cl

99% copper (Pradenas, Zúñiga & Parada, 2006). In the refining and casting stage, the product of the conversion is again purified in a rotary furnace and is then poured into casting wheels, generating 99.7% pure copper, which represents the final product.

The smelted copper concentrate from the conversion stage is transported and fed to the refining furnaces by means of large ladles moved by a bridge crane. Figure 1 is an operations diagram of a refining and casting plant in which the furnaces are next to each casting wheel, providing independent loads. The refining operation is conducted in a set of reactors of different capacity. To load a given refining reactor, it must be in operation, it must not be processing another load, and a certain amount of time must have passed since it was unloaded. The processing time in this stage depends on the content of the material in the furnace and on its chemical characteristics, and therefore, it is known beforehand.

Copper anodes are cast in fixed molds inserted in rotating casting wheels. The molds in a wheel are filled from a pair of refining furnaces next to the casting wheel, but at the time of filling, only one furnace can be operated. The casting wheel turns periodically and sets up other molds to be filled. As the wheel turns, the filled molds are cooled with water until they reach the sector where they are unloaded automatically.

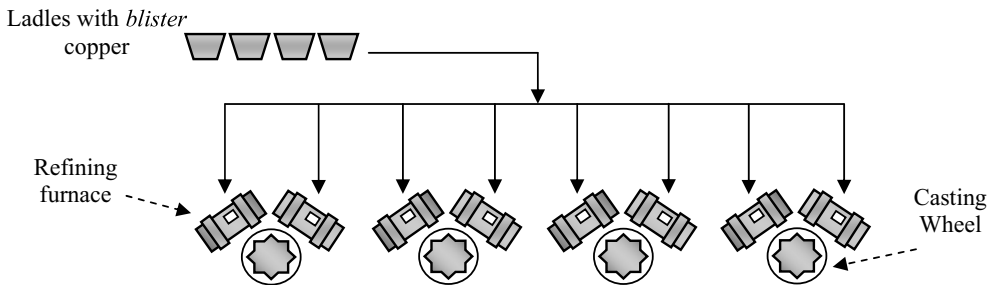


Figure 1 – Configuration of a refining and casting plant.

To get the maximum performance from the equipment involved in the refining and casting process, it is necessary for the operations to be conducted in a coordinated fashion during a planned period, that is, they must be synchronized. These operations include operating the bridge crane, loading the furnaces, running refining cycles, loading the molds, and defining the furnace-mold relation. The complexity of this coordination causes a general tendency to deal with the activity manually, generating, as a result, a series of problems that are translated into inefficiency in the operations. Such inefficiencies include the non-availability of a bridge crane when required, material from the conversion stage that does not find a furnace for loading, and loads in furnaces that do not have an available casting wheel. It is therefore necessary to generate a schedule for loading the refining furnaces in a given planning period (typically one day) that specifies the loading of each ladle with molten material from the refining stage. Also, the order and time at which each refining furnace must feed its corresponding casting wheel must be determined. This gives rise to the problem of the automatic scheduling of operations in the refining and casting stage of a copper concentrate smelter that we call the Refining and casting Scheduling Problem (RCSP). Pradenas *et al.* (2005) developed a constructive heuristic to generate practical solutions

for a copper concentrate smelter. In their work, they mention that the problem can be considered to belong to the family of scheduling problems, but the topic was not explored further. The problem's characteristics place it within the scheduling family of problems, which in general are still a great intellectual challenge because of their NP-hardness in most cases (Pinedo, 2008). In this paper, we propose that RCSP can be transformed into a flexible flow shop problem, one of the most extensively studied cases in recent literature on scheduling, but in this particular case taking into account the particular constraints associated with the metallurgical process (Fattahi, Jolai & Arkat, 2009; Mati, Lahlou & Dauzere-Peres, 2011). The computational difficulties that arise when solving instances of this problem numerically have led to attempts to use various techniques to find good quality approximate solutions (Li, Pan, Suganthan & Chua, 2011; Rajkumar, Asokan, Anilkumar & Page, 2011; Zhang, Gao & Shi, 2011). To approach RCSP, we propose a mathematical model and an algorithm to solve it. We develop a computational experiment considering real operating situations.

Section 2 presents a mathematical model of the refining process with its particular constraints. In Section 3, the problem is characterized, a hierarchical approach is established to solve the problem, and an initial solution is improved by simulated annealing in the search for better feasible solutions. Section 4 presents the results and parameterization, and finally, Section 5 gives the main conclusions.

## 2 MATHEMATICAL REPRESENTATION OF A COPPER REFINING AND CASTING PLANT

According to the characteristics of this production system, some elements are seen that allow the problem to be associated with a flexible *flow shop* type configuration (Pinedo, 2008):

- A job that enters the system has the possibility of continuing at any of the available work centers, *i.e.*, for each stage of the process a job can be processed on any of the machines available at the job centers, characterizing a flexible type configuration (Bagchi, Gupta & Sriskandarajah, 2006).
- When a job's process is finished in the refining furnace, it takes a *unidirectional flow* toward the adjacent casting wheel.
- The machines available at each stage of the process are identical, as they have the same processing speed.

The mathematical model for the problem is constructed with binary integer variables and continuous variables following the strategy that has typically been used to represent scheduling problems (Blazewicz, Dror & Weglarz, 1991). The model is presented in equations (1-10).

### Parameters

The model's parameters are obtained from the unloading of the reactors of the previous stage as well as from the operational characteristics of the refining and casting plant. There are  $n$  jobs

to be processed at  $q$  job centers. Parameter  $p_i$  corresponds to the refining time of job  $i$ , which is a group of ladles with blister copper (in a range of 4 to 6 ladles) that are loaded in a refining furnace. The refining time depends mainly on the concentrate's composition. The arrival time  $r_i$  is obtained from estimates of when the loads will come from the previous conversion stage and considers the time required to transfer the ladles that determine job  $i$  to the refining furnace. The casting time  $c_i$  depends mainly on the amount of loaded material. Parameter  $S$  represents the time required to prepare the wheel before starting a new casting, and  $A$  represents the cleaning time of a refining furnace.

- $p_i$  = refining time of job  $i$ ,
- $c_i$  = casting time of job  $i$ ,
- $r_i$  = time of release or arrival of job  $i$ ,
- $S$  = preparation time of the casting wheel,
- $A$  = preparation time of the furnace after a casting,
- $ENC$  = number of allowed linkages.

### Decision variables

The decision variables used in the model belong to three groups, some binary, some continuous, and some integer. These variables are related to each other and were selected for convenient formulation of the operating constraints. The binary variable  $x_{ij}^k$  takes a value of 1 when job  $i$  is the  $j^{\text{th}}$  job processed on machine  $k$ . In this way each job assigned to a machine has a route and a position  $j$  that delivers the sequence in which it will be processed. Variable  $t_j^k$  denotes the time at which the process of the  $j^{\text{th}}$  job of the sequence corresponding to machine  $k$  begins. Variable  $m_k$  corresponds to the number of jobs assigned to machine  $k$ .

- $x_{ij}^k$  = 1 if job  $i$  is the  $j^{\text{th}}$  job processed on machine  $k$ , and 0 in any other case;
- $t_j^k$  = starting time of the  $j^{\text{th}}$  job on machine  $k$ ;
- $m_k$  = number of jobs assigned to machine  $k$ .

### Auxiliary variables

$y_{jl}^k$  = 1 if  $j = l$  (linkage of the jobs processed on  $k$  and  $k + 1$ ) and 0 in any other case. Linkage occurs when a wheel finishes processing a job at the same instant at which one of the associated furnaces finishes the refining process; then, it is possible to load the wheel again without any prior preparation. The number of linkages per wheel in the real problem is limited because the workers are exposed to elevated temperatures, causing them to become physically exhausted. For that reason, only one linkage per wheel is generally allowed in a daily schedule.

### Objective function and constraints

The function to be minimized (1) in the model corresponds to the makespan of the process, *i.e.*, the time required to complete all the jobs in the system.

Constraint (2) establishes that for every machine there is a set of jobs  $m_k$  and that their sum is equal to the total number of jobs ( $n$ ) to be processed in the system. The group of constraints (3) states that a job does not start its process if it has not arrived at the refining stage ( $r_i$ ), causing the time at which job  $i$  that is in the  $j^{\text{th}}$  position on machine  $k$  starts being processed to be greater than or equal to the time of its arrival. Constraint (4) forces all the jobs to be processed in some machine  $k$  and in a given position  $j$ . Constraint (5) forces at least one of the jobs to be processed in the  $j^{\text{th}}$  position on machine  $k$ . Constraint (6) restricts the sequence of jobs on one machine, preventing them from interrupting each other. Constraint (7) allows a linkage to be produced in a wheel when a particular condition is fulfilled. If that condition is not fulfilled, constraint (8) allows a new process to be started on the machine after the processing of the previous job has been finished and the corresponding casting wheel has been cleaned. Constraint (9) limits the number of linkages that can be produced in the scheduling of a wheel. Finally, constraint (10) shows the general definition of the model's variables.

The general precedence constraints are forced by the entry constraints of the jobs to the furnaces and by the double use of the wheel by contiguous furnaces. Under normal operating conditions, a typical refining and casting plant has 4 conversion reactors, 6 refining furnaces, and three casting wheels that maintain 20 hours of effective operation daily. With these facilities, a plant processes approximately 72 blister copper ladles consolidated in 12 jobs, so the possible routing combinations of the jobs are on the order of  $6^{12}$  and the operating constraints number 948.

$$\text{Min } Z = \text{Max}_{l < k < m} \left\{ \sum_j^{m_k-1} \sum_i^n x_{ij}^k \cdot (t_{j+1}^k - t_j^k) + \sum_i^n k_{im_k}^k \cdot (p_i + c_i) \right\} \quad (1)$$

s.t.

$$\sum_{k=1}^q m_k = n \quad (2)$$

$$t_j^k - \sum_{i=1}^n x_{ij}^k \cdot r \geq 0 \quad j = 1, \dots, m_k \quad k = 1, \dots, q \quad (3)$$

$$\sum_{k=1}^q \sum_{j=1}^{m_k} x_{ij}^k = 1 \quad i = 1, \dots, n \quad (4)$$

$$\sum_{i=1}^n x_{ij}^k \leq 1 \quad j = 1, \dots, m_k \quad k = 1, \dots, q \quad (5)$$

$$t_{j+1}^k \geq t_j^k + \sum_{i=1}^n x_{ij}^k \cdot (p_i + c_i + A) \quad j = 1, \dots, m_k - 1 \quad k = 1, \dots, q \quad (6)$$

$$t_l^{k+1} \geq t_j^k + \sum_{i=1}^n x_{ij}^k \cdot (p_i + c_i) - \sum_{i=1}^n x_{il}^{k+1} \cdot p_i - H \cdot (1 - y_{lj}^k) \quad l, j = 1, \dots, m_k - 1$$

$$k = 1, 3, 5, \dots, q - 1 \quad (7)$$

$$t_l^{k+1} \geq t_j^k + \sum_{i=1}^n x_{ij}^k \cdot (p_i + c_i) + S - H \cdot y_{lj}^k \quad l, j = 1, \dots, m_k - 1$$

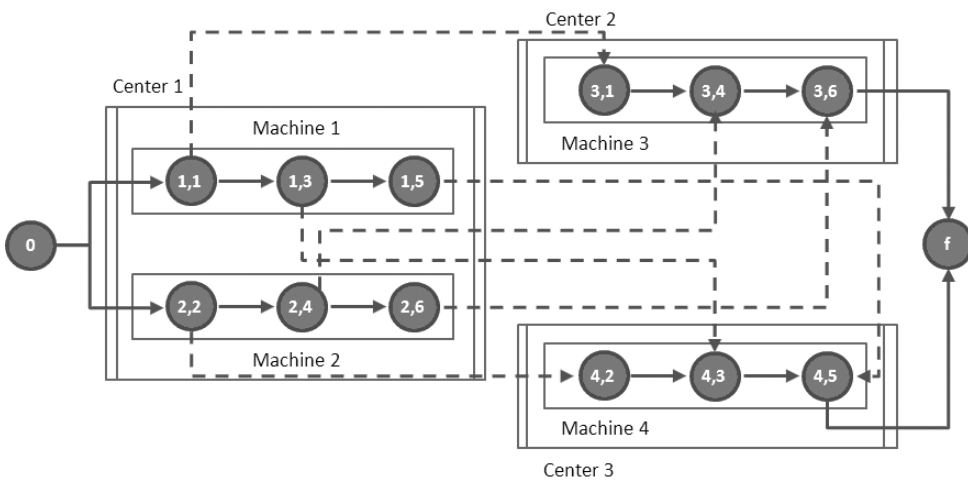
$$k = 1, 3, 5, \dots, q - 1 \tag{8}$$

$$\sum_{l=1}^{m_k} \sum_{j=1}^{m_k} (y_{lj}^k + y_{lj}^{k+1}) \leq ENC \quad i = 1, \dots, n \tag{9}$$

$$t_j^k \geq 0 \quad x_{ij}^k, y_{lj}^k \in \{0, 1\} \quad H \gg \gg 0 \tag{10}$$

**Representation of the problem through a graph**

Typically, a flow shop problem is represented with a directed graph (Balas, 1969; Nowicki & Smutnicki, 1998). However, in this case, two particularities of the problem must be taken into account: the concept of *batch* or *virtual batch* and the *job processing order*. Starting from the unloading of the conversion reactors, the copper ladles are grouped (4 to 6 ladles) into jobs that enter the refining and casting stages associated with their corresponding unloading at different arrival times. Thus, a job constitutes a virtual processing lot, *i.e.*, the jobs are not necessarily available and stored physically one after another in a space or *buffer* of the plant, but rather, the purpose of their grouping is to allow the definition of virtual job sub-batches before each processing center (routing – sequencing), allowing the scheduling problem to be decomposed at the level of each center, as represented schematically in Figure 2.



**Figure 2** – Directed graph  $G(14, 3, 6)$  with 6 jobs and 3 processing centers.

**3 SOLUTION PROCEDURES**

Given a lot of  $N$  jobs to be processed in the refining stage, a batch or subplot is defined as a subset of jobs ( $i$ ) assigned to a center that can be exchangeable among the different centers because the

machines are identical; thus, a batch is not associated with a specific center. Each center  $l$  has a set of jobs  $N_l$  of size  $n_l$ . The processing order of the jobs of each batch  $N_l$  can be expressed as a permutation  $\Pi_l = \{\Pi_l(1), \Pi_l(2), \dots, \Pi_l(n_l)\}$  where  $\Pi_l(p)$  denotes the element belonging to subplot  $l$  that will be processed in position  $p$ . Figure 3 presents a schematic diagram with the methodology used for the refining and casting plant.

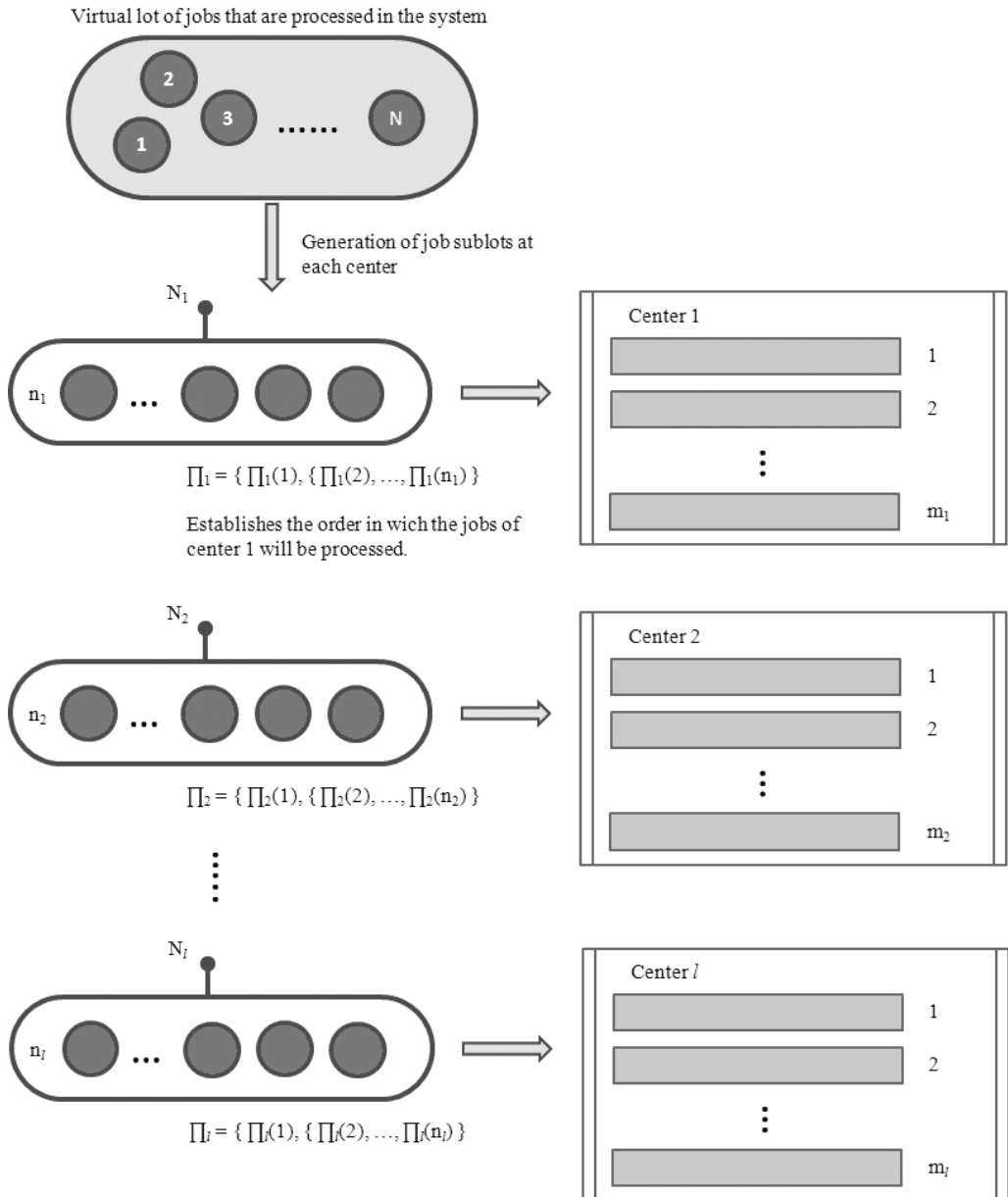


Figure 3 – Scheme of a copper refining and casting plant with batch jobs.

The makespan of the process corresponds to the maximum time for completing a job that is in position  $p$  of subplot  $l$ . Let us consider that each job has an associated furnace-wheel processing time in the job center, so  $p_{li}$  denotes the processing time at center  $l$  of job  $i$ . Therefore, a recursive function can be established for the completion time of a job in a subplot  $C_{1,\Pi l(p)}$ :

$$C_{1,\Pi l(p)} = \begin{cases} \max \{C_{1,\Pi l(p-1)}, a_{\Pi l(p)}\} + p_{l,\Pi l(p)} & \text{Unlinked job} & (11a) \\ \max \{C_{1,\Pi l(p-1)}, r_{\Pi l(p)}, a_{\Pi l(p)}\} + p_{l,\Pi l(p)} & \text{Linked job} & (11b) \end{cases}$$

where,  $p$  is in the discrete set  $\{1, 2, 3, \dots, n_l\}$  and  $C_{1,\Pi l(0)} = 0$ .  $a_{\Pi l(p)}$  is the elapsed time between the output of the conversion stage and the arrival to the refining and casting stage. The process time  $p_{l,\Pi l(p)}$  corresponds to the refining time ( $r$ ) plus casting time ( $m$ ) of the job of subplot  $l$  that is in position  $p$ , *i.e.*:

$$p_{l,\Pi l(p)} = r_{\Pi l(p)} + m_{\Pi l(p)} \tag{12}$$

The completion time of a job for center  $l$  in position  $p$  starts when the previous job is finished (position  $p - 1$  of subplot  $l$ ) provided the job has arrived (position  $p$ ), which is why the maximum of those two times are in equation (11).

$$C_{\max} = \text{Max}\{C_{1,\Pi l(p)}\} \tag{13}$$

Another approach used to solve the FFSP decomposes the problem into two levels: a routing sub-problem and a job shop sub-problem (Brandimarte, 1993). For the routing sub-problem, we developed a job assignment heuristic forming sublots for each processing center starting from a given job sequence. Job sequencing is initially determined with a heuristic based on the LPT dispatch rule (Pinedo, 2008), and at the later stage, simulated annealing (Kirkpatrick Jr & Vecchi, 1983; Talbi, 2009) is used. A diagram of the procedure’s stages is presented in Figure 6. The construction function generates the processing sublots and schedules the subplot jobs inside a center. The sublots are constructed by assigning each job to a processing center according to the procedure presented in Figure 4. On the other hand, to construct the job schedule of a subplot each job is assigned to some machine of the processing center according to the procedure described in Figure 5.

The representation of the subproblem for the simulated annealing method requires the generation of a solution neighboring the current solution. For that purpose, a position  $p$  of the job sequence is chosen randomly, and the neighboring solution is obtained from random exchange among the positions  $p$  by  $p - 1$  or else  $p$  by  $p + 1$ . The new sequence of jobs generates new sublots for each processing center, and the new starting time of each job must be recalculated.

The scheme of Figure 6 describes the general procedure. From the information regarding the available processing centers and the jobs to be processed, an initial (or job entry) sequence to the refining plant is determined, the sublots of jobs for each center are constructed, and the scheduling for each element of the subplot is determined. In this way, the evaluation function calculates the makespan of the process. In the second stage, the interaction between the SA and the generation of sequences takes place. As neighboring job entry sequences are generated,



**Procedure 1: Construction of sublots****Start**

**Given** a sequence of jobs to be processed;

**While** there is some unassigned job in the sequence

**Search** for an available processing center

**If** there is an unoccupied center;

Assign the job to a center

**Otherwise**

Assign the job to the tail of the center that is set free first

**End.**

**Figure 4** – Construction stage of Sublot scheduling.

**Procedure 2: Job scheduling****Start**

**For each** of the sublots associated with the centers;

**While** there is a job of the subplot that has not been scheduled

Choose the first job of the subplot

**Search** for an available machine in the center

**If the center has a machine**

**If** the machine is unoccupied

Schedule the starting time of the job ( $T_i$ )

Schedule the completion time ( $T_f$ )

**Otherwise**

Following  $T_i$  of subplot =  $T_{f-1} + T_{setup\_wheel}$

Schedule its completion time ( $T_f$ )

**Otherwise**

**Choose a machine available at the center**

**If** there is an unoccupied machine

Schedule its starting time of the job ( $T_i$ )

Schedule its completion time ( $T_f$ )

**Otherwise**

**If there is linkage**

Following  $T_i$  of subplot =  $T_{f-1} - T_{refining\ next\ job}$

Schedule its completion time ( $T_f$ )

**If there is no linkage**

$T_i$  following subplot =  $T_{f-1} + T_{setup\_wheel}$

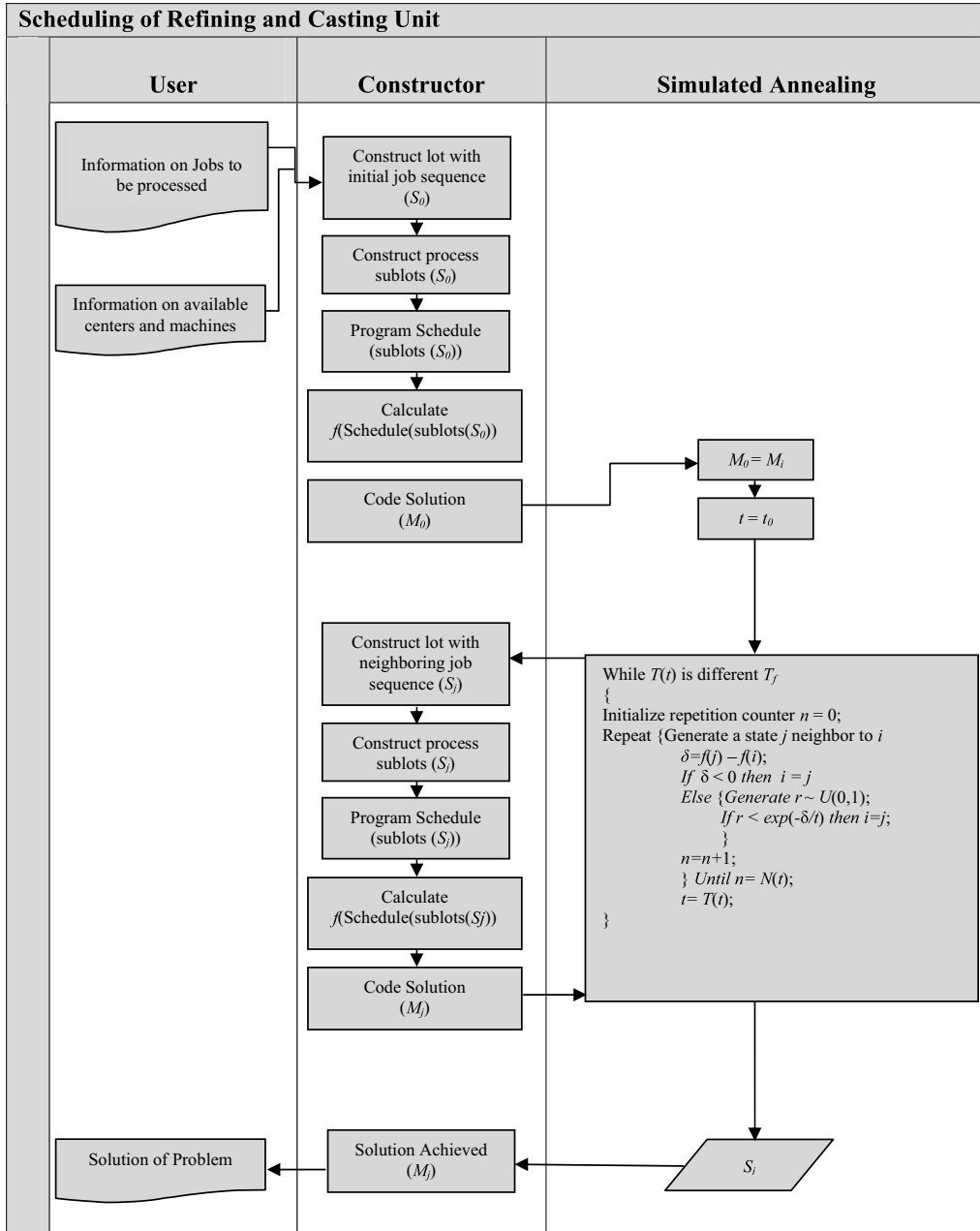
Schedule its completion time ( $T_f$ )

**End while;**

**End for;**

**End.**

**Figure 5** – Construction of subplot scheduling.



**Figure 6** – Interaction between the Construction and Simulated Annealing.

they are entered into the construction module that is in charge of producing the job sublots and scheduling. Then, the evaluation function of the neighboring solution is determined, and it is transmitted to the SA. The process continues until one of the stopping criteria is fulfilled.

## 4 RESULTS

### 4.1 Test problems

To generate the test problems, a computational tool was executed that programs the activities in the conversion plant, which is the previous stage in copper production (Pradenas *et al.*, 2006). The output of this tool produces the input data for the copper refining and casting problem. This information contains data on the instant at which each ladle with molten material is produced in the conversion stage. One RCSP instance is defined by the number of jobs that will be processed, the availability of the processing centers, and the arrival and processing times of each. To generate the test problems, the normal operation of a refining and casting plant is considered, and alterations are made as to the availability of the processing centers and the number of jobs to be processed, increasing the complexity of the real problem instances. Table 1 presents the characteristics of the generated instances and the available centers.

**Table 1** – Characteristics of the jobs.

Problem	No. of instances	Consolidated jobs	Available centers
RM08XX	5	8	3
RM10XX	5	10	3
RM12XX	5	12	3

For each of the five instances of the three types of problems, refining, casting, arrival, and loading times were generated randomly with a uniform triangular probability distribution, as presented in Table 2. Time is measured in minutes and load size in metric tons.

**Table 2** – Distribution of the data that characterize each job.

Data	Distribution function
Refining Time	Uniform [130,175]
Casting Time	Uniform [240,420]
Arrival Time	Uniform [30, 1150]
Load	Triangular [250, 250, 350]

For the studied cases, it is assumed that the processing centers for each type of instance are always available, so that the three casting wheels are always operating and are fed by one of their corresponding refining reactors. Table 3 gives the details.

### Hardware used

All of the programs were executed on a computer with an Intel T2400 – 1.83 GHz processor with 2 Gb of RAM. Both algorithms were implemented in C++ language on a Windows XP operating system.

**Table 3** – Description of the processing centers.

Problem	Center	Available furnace	Furnace capacity
RM08XX	1	1	300
	2	1	300
	3	1	350
RM10XX	1	2	300
	2	1	300
	3	1	350
RM12XX	1	2	300
	2	2	300
	3	1	350

**Parameter definition**

The simulated annealing method has various control parameters whose values must be established so that it will deliver good quality solutions in a reasonable execution time. To achieve an adequate parameterization, use was made of the 15 instances and of the WinCalibra software, whose procedure is based on a *Taguchi Factorial Experiment Design and a local search* (Adenso-Diaz & Laguna, 2006). The values obtained from the parameterization are presented in Table 4.

**Table 4** – Parameter values.

Parameter	Range	Obtained value
Parameter $\alpha$	0.090-0.999	0.995
Initial temperature	1000-1700	1175
Final temperature	0.01-10.00	7.5
Iterations for each temperature level	1-100	76

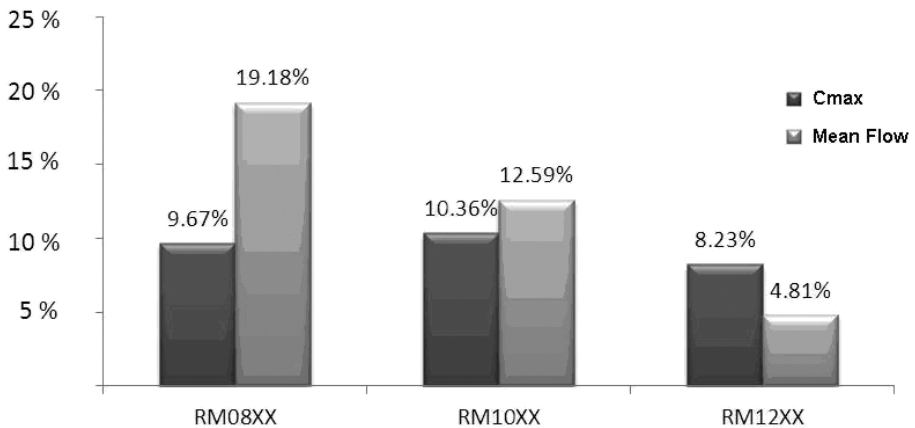
**Schedule generation**

Table 5 shows the results obtained for the 15 instances using the proposed method as well as the constructive heuristic (Pradenas *et al.*, 2005). The completion time ( $C_{max}$ ), the mean flow time, and the CPU time are given for each instance.

The proposed method improves  $C_{max}$  by an average of 9.42% and mean flow time by 12.19% with respect to the constructive heuristic. Even though the CPU time is very low, it is longer in simulated annealing, consistent with the difference between a constructive heuristic and the iterative characteristics of the simulated annealing metaheuristic. Figure 7 shows the average percentage improvement in  $C_{max}$  and mean flow time for each of the three types of problems studied obtained by using the proposed method instead of the constructive heuristic.

**Table 5** – Summary of results.

	Constructive heuristic			Proposed method		
	$C_{max}$ [min]	Mean flow time [min]	CPU time [s]	$C_{max}$ [min]	Mean flow time [min]	CPU time [s]
RM0801	2148	773.13	0.031	1887	554.25	1.266
RM0802	1985	701.00	0.031	1738	579.00	1.204
RM0803	1624	686.63	0.031	1494	599.25	1.219
RM0804	2247	656.13	0.031	2067	541.00	1.187
RM0805	1739	731.38	0.031	1604	585.75	1.203
RM1001	2053	749.70	0.047	1675	747.50	1.218
RM1002	2525	747.00	0.031	2345	692.30	1.265
RM1003	2064	660.20	0.031	1960	643.60	1.235
RM1004	2156	783.90	0.063	1782	502.30	1.234
RM1005	1992	641.10	0.047	1915	532.80	1.234
RM1201	2287	846.42	0.047	2154	800.17	1.265
RM1202	2555	796.92	0.031	2291	734.75	1.281
RM1203	2327	707.50	0.031	2167	692.92	1.281
RM1204	2347	703.25	0.047	2052	624.42	1.234
RM1205	2551	814.33	0.031	2409	834.42	1.25

**Figure 7** – Improvement percentage.

## 5 CONCLUSIONS

In this paper, we have proposed an approach based on the flexible flow shop problem to tackle an operations management problem that arises in a copper smelter in its material refining and casting stage. The problem has additional characteristics beyond those treated in the scheduling literature due to conditions typical of the metallurgical process. To represent the problem, use was made of directed graphs and virtual batch processes of jobs. In this way, an algorithm that operates at two levels is defined. First, from a job sequence constructor, job sublots are configured

for each available refining center in such a way that a route and a sequence for entering the refining center are defined that optimize the use of the casting wheel. At the second level, this constructed solution is improved by means of the simulated annealing method. After solving 15 test problems, the proposed algorithm improved on the *makespan* by 9.42% and the mean flow time by 12.19% with respect to a constructive heuristic.

## ACKNOWLEDGMENTS

Authors were supported by projects: CONICYT: FBO16; UDEC-208.97011-1 and by the Millennium Institute Complex Engineering Systems ICM: P-05-004-F.

## REFERENCES

- [1] ADENSO-DIAZ B & LAGUNA M. 2006. Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research*, **54**(1): 99–114.
- [2] BAGCHI TP, GUPTA JND & SRISKANDARAJAH C. 2006. A review of TSP based approaches for flowshop scheduling. *European Journal of Operational Research*, **169**(3): 816–854.
- [3] BALAS E. 1969. Machine sequencing via disjunctive graphs: an implicit enumeration algorithm. *Operations Research*, **17**(6): 941–957.
- [4] BLAZEWICZ J, DROR M & WEGLARZ J. 1991. Mathematical programming formulations for machine scheduling: a survey. *European Journal of Operational Research*, **51**(3): 283–300.
- [5] BRANDIMARTE P. 1993. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, **41**(3): 157–183.
- [6] FATTAHI P, JOLAI F & ARKAT J. 2009. Flexible job shop scheduling with overlapping in operations. *Applied Mathematical Modelling*, **33**(7): 3076–3087.
- [7] KIRKPATRICK S JR, D GUELAT & VECCHI MP. 1983. Optimization by simulated annealing. *Science*, **220**(4598): 671–680.
- [8] LI J, PAN Q, SUGANTHAN P & CHUA T. 2011. A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. *International Journal of Advanced Manufacturing Technology*, **52**(5-8): 683–697.
- [9] MATI Y, LAHLOU C & DAUZERE-PERES S. 2011. Modelling and solving a practical flexible job-shop scheduling problem with blocking constraints. *International Journal of Production Research*, **49**(8): 2169–2182.
- [10] NOWICKI E & SMUTNICKI C. 1998. The flow shop with parallel machines: A tabu search approach. *European Journal of Operational Research*, **106**(2-3): 226–253.
- [11] PINEDO M. 2008. Scheduling: theory, algorithms, and systems. Springer Verlag.
- [12] PRADENAS L, NÚÑEZ G, PARADA V & FERLAND J. 2005. Gestión de operaciones de refino y Moldeo en la Producción de Cobre. *Revista Ingeniería de Sistemas*, **19**(1): 19–28.
- [13] PRADENAS L, ZÚÑIGA J & PARADA V. 2006. CODELCO, Chile programs its copper-smelting operations. *Interfaces*, **36**(4): 296–301.

- [14] RAJKUMAR M, ASOKAN P, ANILKUMAR N & PAGE T. 2011. A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints. *International Journal of Production Research*, **49**(8): 2409–2423.
- [15] TALBI EG. 2009. *Metaheuristics: From design to implementation*. John Wiley & Sons, Hoboken (NJ).
- [16] ZHANG G, GAO L & SHI Y. 2011. An effective genetic algorithm for the flexible job-shop scheduling problem. *Expert Systems with Applications*, **38**(4): 3563–3573.