Available online at www.sciencedirect.com
ScienceDirect
Int. J. Production Economics 104 (2006) 317–326
international journal of production economics
www.elsevier.com/locate/ijpe

# Production–inventory scheduling using Ant System metaheuristic

Ivan Ferretti, Simone Zanoni*, Lucio Zavanella

*Dipartimento di Ingegneria Meccanica, Facoltà di Ingegneria, Università degli Studi di Brescia, Via Branze, 38, Brescia I-25123, Italy*

Received 4 April 2004; accepted 25 January 2005
Available online 5 March 2005

**Abstract**

The present paper presents the algorithmic solution, based on an Ant System metaheuristic, of an industrial production–inventory problem in a steel continuous-casting plant. The model proposed is based on an objective function, the aim of which is to find the most profitable production schedule of the steel billets. Furthermore, the model takes into account the relevant parameters of the finite-capacity productive system (e.g. set-up and processing times, demand profile, warehouse capacity). Moreover, the make-to-order production environment of the company presents a significant manufacturing phase, which is represented by the billet cooling warehouse (similarly to the drying process in paper and textile production, or maturing in food production): this fact introduces a relevant constraint to production schedule. The study shows the basic criteria used for the problem modelling and the steps proposed for profit optimisation. The Ant System algorithm implemented is discussed and its relevance for the steel plant production management is shown.


*Keywords:* Ant System; Metaheuristics; Scheduling; Continuous casting

## 1. Introduction

As far as the scheduling problem is concerned, several studies highlight the existing gap between theory and practice (e.g. MacCarthy and Liu, 1993), due to the difficulties in the mathematical description of complex industrial contexts.

In the present contribution, a metaheuristic approach, the Ant Colony Optimisation introduced by Dorigo et al. (1991), is applied to the production scheduling of a real industrial context, where the constraint introduced by the finite product warehouse is particularly relevant.

This problem has been discussed, for the first time, by Zanoni and Zavanella (2005), and it has been approached by a Mixed Integer Linear Model (MIP), implemented and solved using common linear optimisation solver (CPLEX

*Corresponding author.
*E-mail address:* zanoni@ing.unibs.it (S. Zanoni).

0925-5273/$ - see front matter 
doi:10.1016/j.ijpe.2005.01.008

6.6.0). For the specific industrial case, the linear model required 8960 integer variables (2240 binary) and 9176 constraints.

Frequently, a linear-programming approach requires a significant effort in the constraint definitions, so as to guarantee their ability in describing the real systems. Therefore, the problem features and conditions are frequently simplified, thus obtaining a mathematical model far from the reality examined. It is easy to understand why, in the industrial context, it is often necessary to abandon the linear programming approach in favour of alternative solution methods.

Between the several methodologies available for approaching and solving complex problems, heuristic algorithms play a significant role. They do not guarantee optimality, but they are fast and, often, successfully applied to hard computational problems (Reeves, 1995). In the present study, the metaheuristic defined as Ant System is adopted: it pertains to the class of the Ant Colony Optimization (ACO) metaheuristics, based on a strategy of solution search derived from the observation of ant behaviour.

It should be highlighted that the present work does not aim to evaluate the outperforming heuristic, among the several available in literature, but it verifies the applicability of an ACO algorithm to an industrial environment, comparing it to the possible MPL solution. The choice of the ACO metaheuristic class has been driven by the promising results of their application to real industrial cases, as presented in literature (Gravel et al., 2002; Gagné et al., 2002). The industrial case discussed may be compared to small TSP problems (10–80 nodes): in such a situation, some authors (Dorigo et al., 1991; Dorigo and Gambarella, 1997; Stützle and Hoos, 1998) show how ACO metaheuristics yield better solutions than other heuristics, also when compared to widely studied ones, as Genetic Algorithms (Gravel et al., 2002; Ying and Liao, 2003).

In their review of solution techniques for scheduling flexible shops, Blazewicz et al. (1996) note that methods such as Simulated Annealing, Tabu Search and Genetic Algorithms are frequently used and they emerge as powerful techniques for this task. Elsewhere, in the scheduling literature, the use of "neural networks" may be found (Huang and Zhang, 1994; Sabuncuoglu and Gurgun, 1996) as well as ACO metaheuristics (Colorni et al., 1994; Stützle, 1998; Den Besten et al., 2000; Ying and Liao, 2003). A large number of similar applications, drawn from various industrial situations, may be found in literature. Franca et al. (1996) use Tabu Search to minimise the makespan of a schedule for parallel processors. Lee and Pinedo (1997) minimise the weighted tardiness in a situation, close to the one discussed here, using a free-phase heuristic and incorporating a Simulated Annealing algorithm. Rubin and Ragatz (1995) use a Genetic Algorithm to schedules $n$ jobs on one machine, so that total tardiness is minimised where set-ups are sequence dependent. Tang et al. (2002) enhanced a linear mixed integer model (Tang et al., 2000) using a Genetic Algorithm for the slab stack-shuffling problem when implementing steel rolling schedules.

The utilisation of mathematical methods for production optimisation in steel plants (the industrial problem we are focusing on) is not new: Tibor (1958) introduced a linear mathematical model to optimise the mix of raw materials and energy consumption. Several contributions followed, involving mathematical programming of the same problem. A survey on this topic (Tang et al., 2001) shows how proposed models refer to the optimisation of the typical stages of steel production, i.e. steel making (SM), continuous casting (CC) and hot rolling (HR). The optimisation may be carried out for the whole system or a part of it. The present contribution is focused on the CC phase, the most relevant phase of mini-steel plants (the final products are billets, differing in section, length and steel composition). Until now, research on production planning and scheduling in steel plants used different kinds of heuristic solution methods, such as *Artificial Intelligence, Tabu Search, Genetic Algorithm, Simulated Annealing and Multi agent methods*.

It is necessary to emphasise that the production scheduling problem of mini-steel plants presents several aspects of originality and interest. This is particularly true for the constraint introduced by the warehouse finished products, which exerts an impact on the productive cycle, being also a part of it, as a consequence of the cooling phase.

## 2. The Ant System

ACO metaheuristics (Colorni et al., 1994; Stützle, 1998) have been recently developed for combinatorial optimisation, inspired by the study of the ant behaviour (Goss et al., 1989). Ants communicate among themselves through pheromone, a substance that they deposit on the ground in variable amounts, as they move. It has been observed that the more ants use a particular path, the more pheromone is deposited on that path and the more it becomes attractive to other ants seeking food. If an obstacle is suddenly placed on an established path, leading to a food source, ants will initially go right or left in a seemingly random manner, but those choosing the shorter path will reach the food more quickly, and they will take the return journey earlier. Therefore, the pheromone on the shorter path will be more strongly reinforced and this path will eventually become the preferred route for the stream of ants (autocatalytic process). These properties lead to the natural application of ACO metaheuristics to the Travelling Salesman Problem (TSP) (e.g. Colorni et al., 1991; Dorigo et al., 1991).

Given a set of $n$ towns, the TSP problem may be stated as the problem of finding the minimal length of a closed tour visiting each town once. We define $d_{ij}$ as the length of the path between town $i$ and town $j$. An instance of the TSP problem is given by a weighted graph $(N, E)$, where $N$ is the set of towns and $E$ is the set of edges between towns, weighted by the distances. At time $t$, the $k$th ant at node $i$ chooses the next node $j$ to visit using the following probabilistic rule $p_{ij}^k(t)$:

$$p_{ij}^k(t) = \begin{cases} \sum\limits_{w \notin \mathrm{tabu}_k} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{[\tau_{iw}(t)]^\alpha [\eta_{iw}]^\beta} & \text{if } j \notin \mathrm{tabu}_k, \\ 0 & \text{if } j \in \mathrm{tabu}_k, \end{cases} \tag{1}$$

where $\tau_{ij}(t)$ is the intensity of trail on edge $(i,j)$ at time $t$; $\eta_{ij} = 1/d_{ij}$ is the "visibility"; $\alpha$ is the relative importance of the edge; $\beta$ is the relative importance of the "visibility"; $\mathrm{tabu}_k$ is a vector, containing the tabu list of the $k$th ant and keeping memory of the towns already visited up to time $t$. This vector avoids ants visiting towns twice, before a tour has been completed.

Therefore, the transition probability represents a compromise between visibility (the closer the town, the higher the probability to choose it) and trail intensity (the higher the traffic on the edge $(i,j)$, the higher its attractiveness, as in Colorni et al. (1991) and Dorigo et al. (1991)).

So as to guarantee the formulation of a feasible tour, nodes already visited are excluded from the choice by the use of the tabu list mentioned. Each ant $k$ will refer to its own tabu list, i.e. $\mathrm{tabu}_k$, containing the ordered list of the nodes already visited. At any given time, several ants seek for a feasible tour: a cycle ends when each ant has completed the tour of the $n$ nodes. In this paper, the algorithm proposed updates the pheromone trail intensity at the end of each cycle: this fact allows us to update the trail according to the evaluation of the solution found in the cycle. The updating rule is implemented as follows:

$$\tau_{ij}(t+n) = \rho\tau_{ij}(t) + \Delta\tau_{ij}, \tag{2}$$

where $\tau_{ij}(t+n)$ is the pheromone trail at the end of cycle; $\rho$ is a coefficient, such that $(1-\rho)$ represents the trail evaporation; $\Delta\tau_{ij}$ is the quantity of trail substance deposited on edge $(i,j)$; $\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$; $\Delta\tau_{ij}^k$ is the quantity of pheromone deposited on edge $(i,j)$ by the $k$th ant;

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & \text{if } (i,j) \in \text{tour done by ant } k, \\ 0 & \text{otherwise;} \end{cases}$$

$Q$ is a constant that represents the total quantity of trail left by an ant during one cycle; $L_k$ is the length of the tour followed by the $k$th ant.

Let $L_k$ be the length of the tour followed at time $t$ by the $k$th ant. This tour length will in turn influence $\Delta\tau_{ij}^k$, by the amount of the pheromone that is added by the $k$th ant to the edge $(i,j)$. This quantity is proportional to the "quality" of the tour, as measured by $Q/L_k$ parameter. The updating of the trail is also influenced by the evaporation factor $(1-\rho)$, which decreases the quantity of the pheromone present on the trail at the previous cycle. Fig. 1 describes the steps of

1. /* Initialization */
**Set** $t := 0$
**For** each edge $(i, j)$ **do**
**set** an initial value $\Delta\tau_{ij}^{k}(t) = 0$
**End for**
**Place** $b_i(t)$ ants on every node $i$
**Set** $s := 1$
**For** $i := 1$ **to** $n$ **do**
**For** $k := 1$ **to** $b_i(t)$ **do**
$tabu_k(s) := i$;
**End for**
**End for**
2. /* Main loop*/
**Repeat**
2.0 **Set** $s := s+1$
2.1 **For** $i := 1$ **to** $n$ **do**
**For** $k := 1$ **to** $b_i(t)$ **do**
*Choose the town j to move to with probability* $p_{ij}^{k}(t)$;
*Move the ant k to j*;
*Insert node j in* $tabu_k(s)$;
**End for**
**End for**
**Until** (*tabu list* is full)
**For** $k := 1$ **to** $m$ **do**
*Move the ant k to start node*;
**End for**
**For** $k := 1$ **to** $m$ **do**
*Compute* $L^k$;
**For** $s := 1$ **to** $n+1$ **do**
**Set** $(h+1) := tabu_k(s),\ tabu_k(s+1)$;
**Set** $\Delta\tau_{hl}(t) \leftarrow \Delta\tau_{hl}(t) + \frac{Q}{L^k}$;
**End for**
**End for**
2.2 **For** (each edge $(i, j)$)
*Compute* $\tau_{ij}(t)$;
**End for**
3. /* End */
*Memorize the shortest tour found up to now*;
**If** ($NC < NC_{MAX}$) **or** (not all the ants choose the same tour)
**then**
/* $NC_{MAX}$ is the number of cycle*/
*Empty all tabu lists*;
**Set** $s := 1$;
**For** $i := 1$ **to** $n$ **do**
**For** $k := 1$ **to** $b_i(t)$ **do**
$tabu_k(s) := i$;
**End for**
**End for**
**Set** $t := t + n$
**For** (each edge $(i, j)$)
**Set** $\Delta\tau_{ij}^{k}(t) = 0$;
**End for**
**Goto** step 2
**Else**
*Print shortest tour*;
*Stop*;
**End if**

Fig. 1. Pseudo-code of the Ant System.

the Ant System proposed by Colorni et al. (1991) and used as reference in the present work.

## 3. The problem description

This study focuses on the optimisation of the production schedule, i.e. the optimal production sequence of the billets, in a "mini steel plant", where the finished product warehouse is a part of the production process. In this area, billets are cooled before their delivery to the customer. Billets differ in steel composition and size (section and length).

The final result of the algorithm is the sequence of customer orders to be produced, thus defining the starting date of each job and considering the delivery date required by the customer. If a delay in the delivery is to be introduced, the objective function accounts it as a penalty cost.

The productive system under consideration is composed by two consecutive phases: the first one is named continuous casting (CC), whose core is an electric arc furnace (EAF), and the second phase requires billet cooling, as carried out in the warehouse. CC process carries out without interruptions, up to the complete emptying of the ladle. Once a ladle is emptied, it must be quickly replaced by a full one, with a minimum loss of time. This peculiarity of the productive cycle imposes that interruptions in billet production must be reduced at the minimum possible, establishing a significant constraint for production planning and scheduling activity. The production process may be summarised as follows: (1) preparation of the iron scraps and alloy elements (EAF loading); (2) fusion; (3) CC of the billets, (4) billet cooling. Billet lengths range from 1.6 to 12 m; sections are square, with sides of 100, 115, 120, 130, 140 and 160 mm. Different compositions of steel may be produced (approximately, one hundred types). Production is continuous on three shifts per day, over the whole week, for a total of 330 working days per year. The EAF capacity is equal to 65 tons, with a 6200 mm diameter. The maximum production is estimated to be 1600–1700 tons per day, i.e. a potential output of about 600,000 tons per year. The warehouse for billet storing and cooling is a $57 \times 26\,\mathrm{m}^2$ area, where an overhead crane is equipped with an electromagnet to transport billets. The warehouse size, further reduced by the billet handling device, represents an important constraint (billets cannot be stocked anywhere else). At present, if customer orders are small (i.e., few casts), billets are arranged in a single-layer stack, thus cooling rapidly ($<12$ hours) to allow truck or railway transportation. If an order is large (i.e., several casts), billets are generally stocked in multiple-layer stacks and cooling times increase (roughly, 12 hours more for each layer added). Of course, there exists a limit to the maximum number of layers, depending on the length and side of the billets (the absolute maximum is 13 layers, for billets 115 mm side and 2 m long).

## 4. Model formulation

In Section 2, the mechanism of the Ant System has been shown as applied to the TSP. In particular, it is possible to define a distance matrix, the elements of which represent the length of the arcs connecting the nodes of the problem graph. This matrix allows the definition of the graph of the specific TSP problem. In the billet scheduling problem, the nodes of the graph do not represent cities, but customer orders that the company received and must delivery in the time horizon fixed for the production plan. Generally, an order may require several casts, one for each type of billet produced (the maximum quantity of billets produced with one cast is 65 tons). The arc length represents the set-up time required to switch from one production batch to another one: batches may differ both in billet section and material composition. This parallelism between TSP and scheduling problem is not new (e.g. Karg and Thompson, 1964). As far as the Ant System is concerned, few applications to real industrial cases are known, with the exception of the one offered by Gravel et al. (2002), where this type of metaheuristic is applied to the scheduling problem of an aluminium smelter plant. The authors found that the Ant metaheuristic gave much better quality results, in much shorter computing times, than their previous Genetic Algorithms. However, Gravel et al.'s problem significantly differs from the case investigated, because of the warehouse constraint. Furthermore, the formulation of the present problem introduces a profit function (PF) encompassing revenue maximisation (i.e., sold billets) and cost minimisation (i.e., billet stocking and delays in order deliveries). The PF is defined as follows:

$$\mathrm{PF} = \sum_{t=1}^{T} [p_\mathrm{r} \cdot S(t) - I_\mathrm{c} \cdot I(t)] - \sum_{i=1}^{n} c_\mathrm{d} \cdot \Delta TS_i \cdot O_i, \tag{3}$$

where $t = 1, \ldots, T$ is the day index; $i = 1, \ldots, n$ is the order index; $p_\mathrm{r}$ is the average profit of one single cast; $S(t)$ is the number of casts sold in day $t$; $I_\mathrm{c}$ is the average holding cost in the ware house per day and per ton of cast steel; $I(t)$ is the number of casts stocked in the warehouse during the $t$th day; $c_\mathrm{d}$ is the daily penalty cost for orders not fulfilled on time; $\Delta TS_i$ is the delay, for the $i$th order, between the shipment date and the date due by contract (null if the shipment occurs on time or in advance); $O_i$ is the number of casts necessary to complete the $i$th order.

The minimisation of the set-up times is not explicitly included in the PF (as in Zanoni and Zavanella, 2005): the lower the set-up times the higher the time available for production, thus obtaining a larger profit. On its turn, the choice of the next node (customer order) the agent (ant) moves to, is influenced by the set-up time. The information concerning distances between graph nodes, and the associated set-up times, is defined as visibility $((t_{\mathrm{SETUP}}(i,j))^{-1})$, where $i$ and $j$ represent, respectively, the $i$th and the $j$th order): if this is the unique strategy used for the node choice a sub-optimal solution is obtained. In the ACO metaheuristic, the visibility information is to be integrated into the pheromone trace deposited by agents (ants) on an arc of the graph. In particular, the definitive trace is left when the sequence has been defined (off-line updating) and the released quantity is related to the goodness of the solution obtained. In the case under examination, the pheromone trace is directly linked to the profit generated by the sequence, evaluated as following:

$$\tau_{ij}(\mathrm{NC}) = \rho\tau_{ij}(\mathrm{NC}-1) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}, \tag{4}$$

where NC is the number of the cycle under evaluation; $\tau_{ij}(\mathrm{NC})$ is the quantity of pheromone on the trail at the end of the cycle; $\tau_{ij}(\mathrm{NC}-1)$ is the current quantity of pheromone on the trail; $\Delta\tau_{ij}^{k}$ is the quantity of trail substance deposited on edge $(i,j)$ by the $k$th ant;

$$\Delta\tau_{ij}^{k} = \begin{cases} \mathrm{PF}_k & \text{if } (i,j) \in \text{ tour done by ant } k, \\ 0 & \text{otherwise;} \end{cases}$$

$\mathrm{PF}_k$ is the profit function value generated by the $k$th ant and the sequence under evaluation.

Moreover, it should be highlighted that, in the Ant System applied to the TSP, the tour generated is closed (i.e. the ant returns to the starting city at the end of its tour). On the contrary, in the present problem, the sequence generated is an open one. Table 1 shows a comparison between the Ant System applied to the TSP instance and to the case under examination.

As far as the billet arrangement in the warehouse is concerned, the billets pertaining to a specific order are not available for the delivery until they are completely cooled. Billet cooling depends on the order size and on the arrangement of the billet layers, which is another decisions variable. This articulated situation is modelled by a sub-algorithm working both in the local search and in the global optimisation. In local search, agents pertaining to the graph choose the next node to reach: the aim of the sub-algorithm it is to appreciate if the order selected could be stocked in the warehouse or not. Thus, the objective of the sub-algorithm is to find the best billet disposition, according to the space available and the minimisation of the cooling time. As in Zanoni and Zavanella (2005), a multiplier coefficient $\delta$, named cooling coefficient and ranging between 0 and 1, has been introduced in the sub-algorithm and used as a variable. Thanks to $\delta$ coefficient, it is possible to evaluate the number of layers in each billet stack according to the following relationship:

$$n^{\circ}\mathrm{layer}_{i\delta} = \lceil \delta \cdot n^{\circ}\max\mathrm{layer}_i \rceil,$$

where $n^{\circ}\mathrm{maxlayer}_i$ sets the maximum number of layers for the stack of the billet type $i$.

As Fig. 2 shows, the development of each ant path occurs according to the probability function $p_{ij}^{k}(t)$. Nodes (i.e., orders) evaluated by the probability function could be chosen from the nodes which do not pertain to the $\mathrm{tabu}_k$ set of the $k$th ant, if space is available in the warehouse for billet stocking. The latter condition implies the selection of a specific billet stack arrangement, depending on $\delta$ parameter, to evaluate the available space as follows:

$$I(t) = I(t-1) + I_{\mathrm{in}}(t) - I_{\mathrm{out}}(t), \tag{5}$$

where $I(t)$ is the number of billet casts stored in the warehouse during the $t$th day; $I_{\mathrm{in}}(t)$ is the number of billet casts produced and stocked in the warehouse in the $t$th day; $I_{\mathrm{out}}(t)$ is the number of billet casts picked up from the warehouse for delivery in the $t$th day.

The Ant System algorithm allows the evaluation of a convenient sequence of orders, taking into account the whole set of billet orders and their

Table 1
Parallelism between TSP Ant System and the billet Ant System

| Features | TSP Ant System | The billets Ant System |
|---|---|---|
| Node | City | Order |
| Distance between node | Length | Set-up time |
| Local search | Select the nearest city | Select the order with lowest set-up time |
| Global optimisation | Updating of the best tour | Updating of the best sequence |
| Objective function | Find the shortest path that links each node | Find the production sequence that maximises the profit |
| Solution generated | Closed sequence | Open sequence |

```
1 /* Initialization */

Read orders data
Write the setup-distance matrix
For each edge (i, j) do
  Set d_ij = t_SETUP(i, j)
  Set τ_ij(0) = τ_0
End for
Place b_i(t') ants (randomly) on every node (order) i
Set s = 1
For δ = 0,1 to 1 step 0,1
  For i = 1 to n do
    For k = 1 to b_i(t') do
      tabu_k(s) = i
Production constrain
      If T_p(O_i) + T_k > T_p then
        split the order quantity O_i
        update T_k and T_p
      else
        update T_k
      end if
Local warehouse constrain
      Choose the optimal billet disposition
      Compute the occupied space
S(t) = S(t-1) + S_in(t) - S_out(t)
      If S(t) > S_MAX then
        the problem is infeasible
        stop (end simulation)
      end if
    end for
  end for
end for

2 /* Main loop */

For δ = 0,1 to 1 step 0,1
  For NC = 1 to NC_MAX
    Repeat
      Set s = s + 1
      For i = 1 to n do
        For k = 1 to b_i(t') do
2.1     Choose the town j to move with probability p^k_ij(s)
        according to equation (1)
        If j ∈ tabu_k(s) then choose an other order j and goto 2.1

Production constrain
          If T_p(O_j) + T_k > T_p then
            split the order quantity O_j
            update T_k and T_p
          else
            update T_k
          end if

Local warehouse constrain
          Choose the optimal billet disposition
          Compute the allocated space S(t)
          If S(t) > S_MAX then
            If order j is the last order between the available then
              the problem is infeasible, eliminate the ant k
            end if
            choose an other order j and goto 2.1
          end if
        end for
      end for
    until (tabu_k is full)

Update pheromone trails
   For each sequence generated by ant k do
    Compute the profit function PF_k
    For each edge (i, j) do
    Update pheromone trails τ_ij(NC) according to equation (4)
End for
   end for
   empty all tabu_k
  end for
end for
choose sequence with the maximum PF between those computed
with different δ values
Print the shortest sequence and its PF
Stop
```

Fig. 2. Pseudo-code of the billets Ant System algorithm.

possible sequences, considering different combinations of parameter $\delta \in [0, 1]$.

The model presented has been implemented in a software, named Ant Colony Optimization Simulator (ACOS). It has been implemented to appreciate the effectiveness of ACO metaheuristic algorithms in the NP-hard optimisation problem: ACOS may solve two different kinds of problem. The first category, i.e. "generic problem", refers to TSP instances of different dimensions: these instances may be solved by the Ant metaheuristic implemented, the Ant System (as in Colorni et al., 1991), the Ant System Rank (as in Bullnheimer et al., 1997), the Ant Colony System (as in Dorigo and Gambarella, 1997) and the Max Min Ant System (as in Stützle and Hoos, 1998). The second category, i.e. "specific problem", is the peculiar application allowing the solution of the problem examined by an ACO metaheuristic. The programme presents a very flexible structure that allows an easy formalisation of the problem and its adjustment for the solution.

## 5. The model evaluation

Firstly, the performances of the solution approach and the metaheuristic have been evaluated comparing it with the mixed integer linear model proposed in Zanoni and Zavanella (2005), i.e. MIP in the remainder. The comparison is quite plain, because of the similarities in terms of variables in the output of the two approaches. The results refer to several experimental simulations, based on different profiles of billets demand: these results (Fig. 3) show how the behaviour of the two models is extremely close, under conditions hereafter discussed. Given a billet order plan, Fig. 3 shows the trend of the profit while varying the constraint of the warehouse length. It should be underlined that the Ant System model didn't find always the optimal solution, but the relative error of its solution ranges between 0.08% and 0.47%.

After this preliminary comparison, additional simulation experiments were carried out: they were grouped into different classes, divided on the basis of the saturation of the plant productive capacity. Fig. 4 shows the profit values: for each saturation level of the plant, 10 different demand profiles were randomly generated, according to the

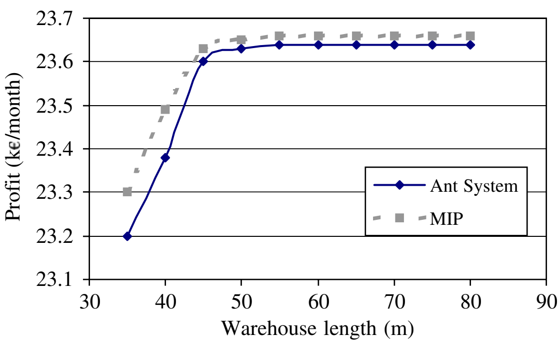


Fig. 3. Trend of the profits with varying warehouse length (€ figures are scaled arbitrarily due to confidentiality reasons).

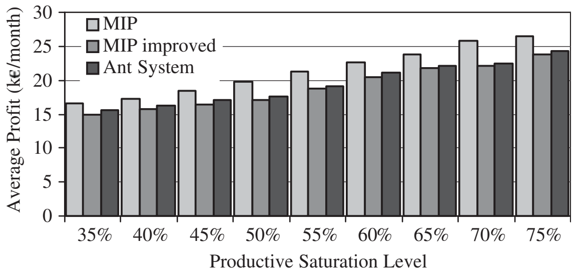


Fig. 4. Comparison between the models (€ figures are scaled arbitrarily due to confidentiality reasons).

historical data of the company. The saturation level is defined as the ratio between the net production time and the total time available for production. Fig. 4 compares the performances of the ACO metaheuristic, as proposed in the present study, the MIP (Zanoni and Zavanella, 2005) and the improved version of the MIP itself. This latter MIP version enhances the former MIP-base solution, so as to introduce some system constraints which could not be included in the Linear Model of the MIP-base itself. The main improvement introduced avoids the order splitting into multiple deliveries (such a condition is unrealistic for the industrial context and it cannot be formulated by means of linear constraints). Furthermore, an algorithmic control imposes the uniqueness of the delivery of one billet order to the customer. This control is included in the Ant System model, too (Fig. 2), thus the comparison between the MIP improved and the Ant System is the most appropriate one. It is evident that the MIP model offers a better solution, but the Ant System model accurately fits the real industrial situation, thanks to its detailed description of the productive system. On the opposite, the constraints slightly differ in the MIP model, being frequently "softer", because of the need for a linear formulation of constraints themselves.

However, a frequent industrial situation refers to a demand profile which requires the saturation of the production system: such a condition is particularly relevant in the industrial case considered. Therefore, other simulation experiments have been carried out and this set of experiments was arranged according to the criticality presented

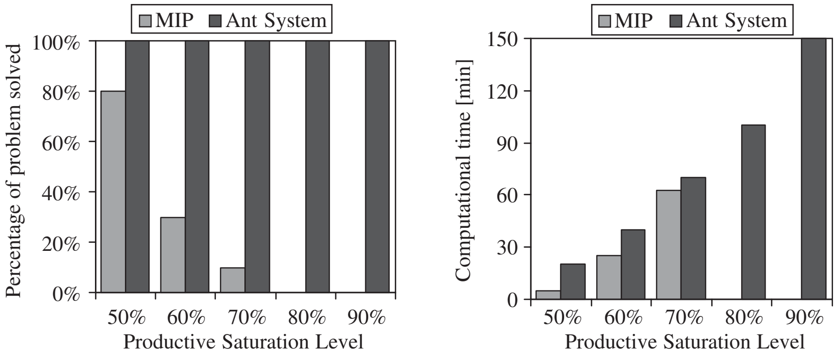


Fig. 5. Evaluation of the model performances while varying the productive capacity.

also in Zanoni and Zavanella (2005) and identified as a weak point of the MIP. This particular situation (high saturation) is also the most relevant and interesting for the industrial case considered: it amplifies the complexity and hardness of this already tricky problem. The results of the simulation experiments are still based on ten different demand profiles, randomly generated according to the historical data of the company. Fig. 5a shows how the MIP model fails in these cases, being unable to find the optimal solution with saturation higher than 70%. However, it should be highlighted that the MIP model finds the solution quite quickly (when it finds it). Results are shown in Fig. 5b, comparing the computational times between the two different approaches adopted. These results reflect the complexity and completeness of the Ant System model, which is extremely close to the real industrial context.

## 6. Conclusions

The present study discussed a model for the solution of a particular problem observed in a real industrial case. The adopted approach is based on a metaheuristic algorithm, the Ant System. In particular, the problem of production planning and scheduling optimisation has been considered for a mini steel plant. In this industrial case, the final product warehouse plays a productive role, being the cooling area of the billets. The study is completed by a set of simulation experiments, developed while using the proposed model (Ant System based) and the reference model (based on a mixed linear approach and discussed in Zanoni and Zavanella, 2005). While developing the Ant System model, a remarkable effort was produced to describe in detail the real condition of the industrial process examined. The results obtained are shown in Section 5 and they show how the approach proposed may provide good solutions in acceptable computation times, thus fulfilling the industrial needs.

## References

Blazewicz, J., Wolfang, D., Pesch, E., 1996. The job shop scheduling problem: Conventional and new solution techniques. European Journal of Operation Research 93, 1–33.

Bullnheimer, B., Hartl, R.F., Strauss, C., 1997. A new rank based version of the Ant System—a computetional study. Proceedings of the Second International Conference on Metaheuristics, Sophia-Antipolis, France.

Colorni, A., Dorigo, M., Maniezzo, V., 1991. Distributed optimization by ant colonies. Proceedings of the European Conference of Artificial Life, pp. 134–142.

Colorni, A., Dorigo, M., Maniezzo, V., 1994. Ant System for job-shop scheduling. Belgian Journal of Operation Research 34, 39–53.

Den Besten, M., Stützle, T., Dorigo, M., 2000. Ant Colony Optimization for the total weighted tardiness problem. Proceedings of the Sixth International Conference Lecture Notes on Computer Science, vol. 191, pp. 611–620.

Dorigo, M., Gambarella, L.M., 1997. Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 53–66.

Dorigo, M., Maniezzo, V., Colorni, A., 1991. Positive feedback as a search strategy. Technical Report no. 91-016, Dipartimento di Elettronica Politecnico di Milano.

Franca, P.M., Gendreau, M., Laporte, G., Muller, F.M., 1996. A tabu search heuristic for the multiprocessor scheduling problem with sequence dependent of setup time. International Journal of Production Economics 43, 79–89.

Gagné, C., Price, W.L., Gravel, M., 2002. Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence dependent setup time. Journal of the Operations Research Society 53, 895–906.

Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M., 1989. Self-organized shortcuts in the Argentine ant. Naturwissenschaften, 579–581.

Gravel, M., Price, W.L., Gagné, C., 2002. Scheduling continuous casting of aluminium using a multiple objective ant colony optimization metaheuristic. European Journal of Operational Research 143, 218–229.

Huang, S.H., Zhang, H.C., 1994. Artificial neural networks in manufacturing: Concepts, applications and perspectives. IEEE Transactions on Components, Packaging and Manufacturing Technology, Part A 212–228.

Karg, R., Thompson, G.L., 1964. A heuristic approach to solving travelling salesman problem. Management Science 10, 225–248.

Lee, H.L., Pinedo, M., 1997. Scheduling jobs on parallel machines with sequence dependent setup time. European Journal of Operational Research 100, 464–474.

MacCarthy, B.L., Liu, J., 1993. Addressing the gap in scheduling research: A review of optimization and heuristic methods in production scheduling. International Journal of Production Research 31, 59–79.

Reeves, C.R., 1995. Modern Heuristic Techniques for Combinatorial Problems. McGraw-Hill, London.

Rubin, P.A., Ragatz, G.L., 1995. Scheduling in a sequence dependent setup environment with genetic search. Computers and Operations Research 22, 85–99.

Sabuncuoglu, I., Gurgun, B., 1996. A neural network model for scheduling problems. European Journal of Operational Research 93, 288–299.

Stützle, T., 1998. An ant approach to the flow shop problem. Proceedings of the Sixth European Congress on Intelligent Techniques and Soft Computing, vol. 3, pp. 1560–1564.

Stützle, T., Hoos, H., 1998. The Max–min Ant System and local search for the traveling salesman problem. In: Voss, S., Martello, S., Osman, I.H., Runcairol, C. (Eds.), Meta-Heuristic: Advances and Trends in Local Search Paradigms for Optimization. Kluwer, Boston, pp. 137–154.

Tang, L., Liu, J., Rong, A., Yang, Z., 2000. A multiple travelling salesman problem model for hot rolling scheduling in Schangai Baoshan Iron and Steel Complex. European Journal of Operational Research 124, 267–282.

Tang, L., Liu, J., Rong, A., Yang, Z., 2001. A review of planning and scheduling systems and methods for integrated steel production. European Journal of Operational Research 133, 1–20.

Tang, L., Liu, J., Rong, A., Yang, Z., 2002. Modelling and a genetic algorithm solution for the slab stack shuffling problem when implementing steel rolling schedules. International Journal of Production Research 40, 1583–1595.

Tibor, F., 1958. A linear programming model of integrated iron and steel production. Management Science 4, 415–449.

Ying, K.C., Liao, C.J., 2003. An Ant colony system approach for scheduling problems. Production Planning and Control 14, 69–75.

Zanoni, S., Zavanella, L., 2005. Model and analysis of integrated production–inventory system: The case of steel production. International Journal of Production Economics 93/94, 197–205.