Solving the Steel Continuous Casting Problem using an Artificial Intelligence Model

Achraf BERRAJAA
INSA Euro-Mediterranean, Euromed Research Center,
Euromed University of Fes, Fes, Morocco

***Abstract*—Over the past decade, the steel continuous casting problem has revolutionized in important and remarkable ways. In this paper, we consider a multiple parallel device for the steel continuous casting problem (SCC) known as one of the hardest scheduling problem. The SCC problem is an important NP-hard combinatorial optimization problem and can be seen as three stages hybrid flowshop problem. We have proposed to solve it a recurrent neural network (RNN) with LSTM cells that we will executed in the cloud. For our problem, we consider several machines at each stage that are the converter stage, the refining stage and the continuous casting stage. We formulate the mathematical model and implemented a RNN with LSTM cells to approximately solve the problem. The proposed neural network has been trained on a big dataSet, which contains 10 000 real use cases and others generated randomly. The performances of the proposed model are very interesting such that the success rate is 93% and able to resolve large instances while the traditional approaches are limited and fail to resolve very large instances. We analyzed the results taking into account the quality of the solution and the prediction time to highlight the performance of the approach.**

Keywords*—*Artificial intelligence; SCC Program; RNN; LSTM; big data

I. Introduction

Iron and steel industry is the cornerstone of an industrialized economy. Since it is capital and energy intensive, companies have constantly laid great emphasis on technological advances to be employed in the production process in order both to increase productivity and to save energy. Due to the complex production process and potential constraints, the latter faces difficult planning and scheduling problems. For example, in the scheduling problem, we usually define a set of n resources, a set of m tasks and a specific optimization goal, called makespan Sol_{max}. The classic flowshop (FS) considers scheduling a set of tasks on one machine at each stage, while the hybrid flowshop (HFS) aims to schedule a flow shop with multiple parallel machines at each stage [1]. In the steelmaking industry, we have three main stages principals for the production that are the converters (CV), the refining stands (RS) and the continuous castings (CC) stages. Each one can include one or more devices, and each product is processed on only one device in each stage. Fig. 1 summarized this configuration, the SCC problem can be seen as a hybrid flow shop. More generally, the goal of SCC is to determine the sequence, timing, and system of equipment involved in the entire production process. This problem is a NP-hard combinatorial optimization and it is considered to be one of the more difficult scheduling problems in the literature [30].

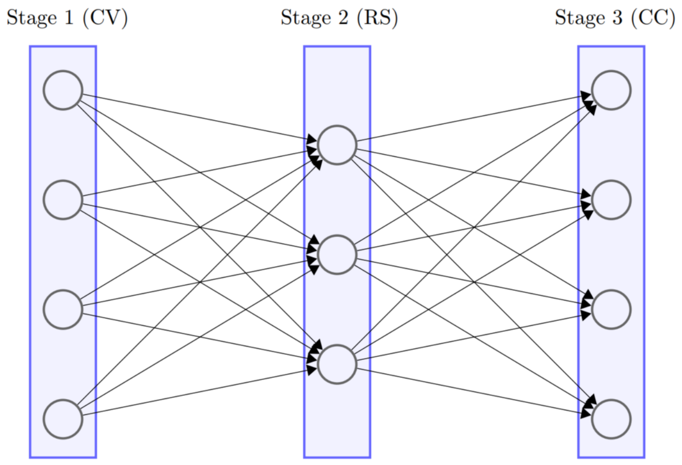

Fig. 1. The Principle Stage of the SCC.

The principal stages and constraints are as follows :

1) The length of stay is minimized,
2) The deadline must be met,
3) The continuity constraint must be satisfied.

We propose in this research, a system with three stage M_i machines at each stage $i(i = 1, \dots, 3)$, such that M_1 identical parallel CV machines are available in the first stage (denoted as CV_{M_1}), M_2 unify the parallel RS machines at stage two (denoted as RS_{M_2}) and M_3 unrelated parallel CC machines at stage three (denoted as CC_{M_3}). In addition, we consider the inter-sequence correlation setting time between every two consecutive sequences λ to be processed on the same CC machine with non-preemptive scheduling. Similarly to the notation by [8], we write our system as a hybrid flowshop (HFS) : $CV_{M_1}, RS_{M_2}, CC_{M_3} \mid \lambda \parallel Sol_{max}$. The first modeling of the problem was considered as a special case of m-stages HFS, which has irrelevant machines and related setup time, which proved to be NP-Hard [14]. For example, [22] has proven the NP hardness of a single machine with a set time.

Some recent research work have been developed to study the steel making continuous casting problem. Most of the developed models deal with high complexity, so solving them as optimal is not always effective, especially for the large instances problems. The most of the methods proposed for the HFS problems are approximation methods, most of which are meta-heuristics. The SCC problem is a special set time scheduling problem, which is difficult to solve due to the high computational complexity [22]. The SCC problem is based on a chronological step chain (the three stages: CV, RS, and CC), where CV, RS, and CC devices are considered to be single, and only one sequence of loads is processed on the machine M_i according to the rules. Among the key rules of the SCC problem, the charging will not start processing on the device until the previous charge is completed. This architecture can be represented by a neural network (forward propagation principle [11]) which allows the automation of the problem SCC. To the best of our knowledge, this is the first work that proposes a model of artificial intelligence to solve the problem of SCC, in particular a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) cells.

In the remainder, the article is organized as follows: Section 2, presents a literature review on the works related to the steelmaking industry in general and the continuous casting in particular. Sections 3 and 4 detail the structure, the sequences and charges, the constraints and formulate the objective function and the mathematical model for the SCC. Section 5 we present the construction of learning big data and we describe our recurrent neural network with LSTM cells for SCC. Section 6 is devoted to the numerical experiments. We also discuss and comment the obtained results and show the efficiency of the method. Finally, Section 7, presents a conclusion that summarizes the study and gives some potential perspectives for the developed approach to improve the current results for the SCC.

II. Related Work and Problem Statement

Continuous casting is one of the most commonly used processes in steel production and has received special attention in the past ten years [31]. Several research works have been developed, both exact and heuristic algorithms have been elaborated to solve several SCC problem variants. The authors in [28] reviewed several SCC models used in steelmaking production, and other works considered mathematical and non-mathematical techniques [3] to solve these variants. Optimization methods such as genetic algorithm (GA) [33], taboo search (TS) [14], mathematical model [32] or swarm intelligence optimization [10] have been developed to optimize production. The hybrid method [27] is also tailored for a continuous steel plant.

Also exact MILP methods are proposed to unravel the SCC. Usually, these methods can only solve small problem instances with commercial solvers, but the real challenge is to use smart and optimized methods to solve large instances. The author in [29] proposed a MILP model for the production order scheduling problem. This model describes a system with more than two machines in each stage, and also more than two sequences must be processed in the final stage, in addition there is an inter-sequence setting time. This model is solved by commercial solvers for relatively small instances. In [9] a mathematical model of the process and a state-of-the-art industrial control system are presented, and they mentioned the importance of using a real-time computational model. In [6], a method has been developed for a steelmaking plant with 2 converters, 2 refining stands, and 2 continuous casters. The authors in [12] develop a discrete-time mixed-integer linear programming (MILP) formulation for a new SCC scheduling problem where different processing routes are used to produce diversified and personalized slab products. The authors in [23] developed a method to minimize the total delay and waste of scheduling problems for different product types by minimizing batches.

Scheduling problems are mainly NP-Hard optimization problems ([5]). In order to overcome their complexity, heuristics and approximation methods may proposed a solution for this types of problems. In the literature approximation methods were developed for SCC. In [18], heuristic-based combinatorial auction technology is also used to solve the SCC containing 4 production orders and a 10-hour planning range. The authors in [16] have developed approximate techniques to solve some planning and scheduling problems of downstream production lines. In [13], the authors have developed a soft-decision based two-layered approach for strong uncertain scheduling. The paper [26] presents an Improved Artificial Bee Colony (IABC) algorithm for the SCC scheduling. Other uncertainty optimizations have been developed for SCC, such as a novel efficient solution algorithm using Augmented Lagrange multiplier method (e-ALM) through relaxation of the coupling constraints and incorporation of penalty components [12].

Several evolutionary algorithms have also been developed for SCC. In [20], we find the description several approaches for computerized scheduling solutions. They include application of techniques in operations research, artificial intelligence, and a hybrid of these two. Nature-inspired optimization methods have also been used in SCC. For example, in [7] the steel industry has been modeled through a combination of a steady-state heat transfer approach and a pareto-converging genetic algorithm (PCGA). In [2], the authors developed a algorithm, that is based on a combination of ant colony optimization and non-linear optimization methods. The authors in [17] have implemented a multi-objective GA to minimize penalties for completion and lateness. In [24], the authors developed a swarm heuristic method for SCC. In the paper of [25], the authors develop a discrete-time mixed-integer linear programming (MILP) formulation for a new SCC scheduling problem where different processing routes are used to produce diversified and personalized slab products. A multi-objective hybrid genetic algorithm combined with local search was presented in [32], in which the enhanced evolutionary mechanisms combined with the improved genetic operators and the local search were also designed. Regarding this work, our approach is based on an artificial neural network, where artificial neural network, is a system whose design is originally schematically inspired by the functioning of biological neurons. The idea has no relation to the way a human being reacts, but to the way of designing the data. It adapts and gives good results when there is a lot of sample information (Features). To the best of our knowledge, this is the first work that proposes a model of artificial intelligence to solve the problem of SCC, in particular a recurrent neural network with long short-term memory cells.

A. Problem Statement

The system processes the costs of different stages under the continuity constraint $(i = 1, \ldots, 3)$. These sequences are preordered costs on the continuous castings machine, and their costs are allocated to any converter machine. Since handling and transportation operations occur in the entire process, the transfer time between stages $\tau_{i,i+1}$ (i = 1, 2) is considered.

We also believe that all converters devices have the earliest and latest start time for the first charge of a sequence with a usable date. The working principle of the system is as follows:

- A sequence represent a set of charges dedicated to one continuous casting machine, and has priority constraints on fees;
- The processing time is limited for any converter, who can be used in the first stage;
- There is a limited residence time (transmission) between the termination of charging in converter and the start of continuous casting to comply with the required temperature;
- Due to continuous constraints, there is no idle time for the charges on the continuous casting;
- The processing time of the continuous casting stage is a decision variable belonging to a given interval;
- The sequence start time at continuous casting is bounded, and the delay between two consecutive sequences is the setup time we define.

The sorted total batch represents the industrial order book so that the sequence can be assigned to the continuous casting.

1) Concept of Sequence and Continuity Constraints: The sequence Seq_{l_3} is defined as a set of $n_{Seq_{l_3}}$ charges to be used in the M_3 casters CC_{l_3} $(l_3 = 1, \dots, M_3)$. The relevant constraints are summarized as follows:

1) The converter can be used in the first stage, there is no specification, and it has a constant processing time and availability date;
2) There is a transition time $\tau_{i,i+1}$ between all stages i (i = 1, 2);
3) The stay (transit) time of the charge j (the time between the termination of the first stage $i = 1$ and the start of the final stage) must not exceed a certain value $T_{l_3,j}$;
4) No idle time is allowed between two consecutive charges in the same sequence in the third stage (continuous constraint);
5) The setting time depends on the sequence time;

III. Continuous Casting of the Ordered Sequences

This section describes all the data (sets, indexes, etc.), parameters, and decision variables that describe the problem being studied. For a given CC_{l_3}, a pre-ordered dedicated sequence Seq_{l_3} $(Seq_{l_3} = 1, \dots, S_{l_3})$ is a list of $n_{Seq_{l_3}}$ charges to continuously process on CC_{l_3} with setup times.

Collections, constants, and indexes :
i : the index of the stage, $i = 1, \dots, 3$;
M_i : the number of the machines l_i at stage i $(l_i = 1, \dots, M_i)$;
S_{l_3} : the number of the sequences Seq_{l_3} to be processed on the machine CC_{l_3};
Seq_{l_3} : a sequence to process on CC_{l_3} $(Seq_{l_3} = 1, \dots, S_{l_3})$;
$n_{Seq_{l_3}} = |Seq_{l_3}|$: the number of charges (jobs) j of the sequence Seq_{l_3};
$n_{l_3} = \sum_{Seq_{l_3}=1}^{S_{l_3}} n_{Seq_{l_3}}$: the total number of the charges to process on CC_{l_3} $(l_3 = 1, \dots, M_3)$;
$n = \sum_{l_3=1}^{M_3} n_{l_3}$: the total number of the charges;
j : the charge index dedicated to the sequence Seq_{l_3} of CC_{l_3}, $j = 1, \dots, n_{l_3}$;
l_1 : the index of the first stage machine CV_{l_1}, with $(l_1 = 1, \dots, M_1)$;

The consider assumptions :
ω_{l_1} : position of a charge on CV_{l_1}, $\omega_{l_1} = 1, \dots, \Pi_{l_1}$;
$l_1 = 1, \dots, M_1 - 1$
$\Pi_{l_1} = \frac{\sum_{l_3=1}^{M_3} n_{l_3}+1}{M_1}$;
ω_{M_1} : position of a charge on the last machine CV_{M_1}, $\omega_{M_1} = 1, \dots, \Pi_{M_1}$;
$\Pi_{M_1} = \frac{\sum_{l_3=1}^{M_3} n_{l_3}}{M_1}$;

Settings and parameters
pro^1 : processing time (constant) for a charge j on any one of the CV_{l_1} at the 1st stage;
$pro^2_{l_2}$: processing time for a charge j on the RS_{l_2};
$[P^{min}_{l_3,j}, P^{max}_{l_3,j}]$: the interval of $pro^3_{l_3,j}$ is dedicated to the processing time of the cost j of CC_{l_3};
$[\lambda^{min}_{Seq_{l_3}}, \lambda^{max}_{Seq_{l_3}}]$: the interval between sequences depends on the setup time $\lambda_{Seq_{l_3}}$.
$date_{l_1}$: available date of CV_{l_1};
$T_{l_3,j}$: maximum allowed sojourn time for a charge j between the termination of any processing in CV_{l_3} and the start of processing in CC_{l_3} $(j = 1, \dots, n_{l_3}$, $l_i = 1, \dots, M_i, i = 1, 3)$;
τ_{12} (resp. τ_{23}) : transfer time required between CV_{l_1} (stage 1) and RS_{l_2} (stage 2) (resp. RS_{l_2} (stage 2) and CC_{l_2} (stage 3)).

Decision variables
The model considers continuous and binary decision variables. For any $l_3 = 1, \dots, M_3$:
$x^i_{l_3,j}$: start time of the charge j dedicated to CC_{l_3} at stage i $(i = 1, 2, 3)$ for $j = 1, \dots, n_{l_3}$,
$pro^3_{l_3,j}$: processing time dedicated to the charge j of CC_{l_3} at the 3rd stage for $j = 1, \dots, n_{l_3}$,
$\lambda_{Seq_{l_3}}$: setup time between two consecutive sequences, $(Seq-1)_{l_3}$ and Seq_{l_3} to process at CC_{l_3}, that occurs between the charge $n_{(Seq-1)_{l_3}}$ and the charge $l_{Seq_{l_3}}$.

$$y^{l_3}_{j,\omega_{l_1}} = \begin{cases} 1 & \text{if charge j dedicated to } CC_{l_3} \\ & \text{is assigned to a position } \omega_{l_1} \text{ in } CV_{l_1} \\ 0 & \text{otherwise} \end{cases}$$

$$j = 1, \dots, n_{l_3}, \omega_{l_1} = 1, \dots, \Pi_{l_1}, l_1 = 1, \dots, M_1.$$

IV. The Mathematical Model for the SCC

In the research problem, we intend to maximize productivity, that is, minimize the manufacturing span (Sol_{max}) and the sequence-dependent setup time, which represents the required duration between two consecutive sequences.

Our modeling method uses the position of the sequence charge and the priority defined by the processing start time of stage 1 (CV). We are also considering pre-orders for charging CC machines in the third stage.

A. The Constraints

We may remember that due to the high complexity of its structure and function, this problem is subject to several constraints that we detailed below:

$$\sum_{l_1=1}^{M_1}\sum_{\omega_{l_1}=1}^{\Pi_{l_1}} y^{l_3}_{j,\omega_{l_1}} = 1;\quad j=1,\ldots,n_{l_3},\quad l_3=1,\ldots,M_3 \qquad (1)$$

$$\sum_{l_3=1}^{M_3}\sum_{j=1}^{n_{l_3}} y^{l_3}_{j,\omega_{l_1}} = 1;\quad \omega_{l_1}=1,\ldots,\Pi_{l_1},\quad l_3=1,\ldots,M_3$$
$$\text{and } \sum_{l_3=1}^{M_3} y^{1}_{1,1} = 1 \qquad (2)$$

$$y^{l_3}_{j+1,t_{(l_1=1)}+1} \le \sum_{l_1=1}^{M_1-1}\sum_{\omega_{l_1}=1}^{t_{l_1}} y^{l_3}_{j,\omega_{l_1}}; j=1,\ldots,n_{l_3},$$
$$t_{l_1}=1,\ldots,\Pi_{l_1}-1 \qquad (3)$$

$$y^{l_3}_{j+1,t_{l_{M_1}}} \le \sum_{l_1=1}^{M_1}\sum_{\omega_{l_1}=1}^{t_{l_1}} y^{l_3}_{j,\omega_{l_1}} - y^{l_3}_{j,t_{l_{M_1}}};\ j=1,\ldots,n_{l_3}-1,$$
$$t_{l_1}=1,\ldots,\Pi_{l_1}; \qquad (4)$$

$$z^1_{l_3,j} = \sum_{l_1=1}^{M_1}\sum_{\omega_{l_1}=1}^{\Pi_{l_1}} (date_{l_1}+pro^1(\omega_{l_1}-1))y^{l_3}_{j,\omega_{l_1}};\ j=1,\ldots,n_{l_3}; \qquad (5)$$

$$z^2_{l_3,j} \ge z^1_{l_3,j} + pro^1 + \tau_{12}; j=1,\ldots,n_{l_3}; \qquad (6)$$

$$z^3_{l_3,j} \ge z^2_{l_3,j} + pro^2_{l_2} + \tau_{23}; j=1,\ldots,n_{l_3}; l_2=1,\ldots,M_2; \qquad (7)$$

$$z^2_{l_3,j+1} \ge z^2_{l_3,j} + pro^2_{l_2}; j=1,\ldots,n_{l_3}-1; l_2=1,\ldots,M_2; \qquad (8)$$

$$z^3_{l_3,j+1} = z^3_{l_3,j} + pro^3_{l_3,j};$$
$$\forall j \notin \{n_1, n_1+n_2,\ldots,\sum_{Seq_{l_3}=1}^{S_{l_3}} n_{Seq_{l_3}}\}, \forall\ l_3=1,\ldots,M_3; \qquad (9)$$

$$z^3_{l_3,j+1} \ge z^3_{l_3,j} + pro^3_{l_3,j} + \lambda_r;$$
$$\forall j = \sum_{Seq_{l_3}=1}^{r} n_{Seq_{l_3}}, r=1,\ldots,S_{l_3}-1; \qquad (10)$$

$$P^{min}_{l_3,j} \le pro^3_{l_3,j} \le P^{max}_{l_3,j}; j=1,\ldots,n_{l_3} \qquad (11)$$

$$z^3_{l_3,j} - (z^1_{l_3,j} + pro^1) \le T_{l_3,j}; j=1,\ldots,n_{l_1} \qquad (12)$$

$$\lambda^{min}_{Seq_{l_3}} \le \lambda_{Seq_{l_3}} \le \lambda^{max}_{Seq_{l_3}}; Seq_{l_3}=1,\ldots,S_{l_3}-1 \qquad (13)$$

The constraint (1) represents that the charge j is allocated to only one converter machine and is located in only one position ω_{l_1}. The constraint (2) represents that the converter CV_{l_1} must process the charge j once and only once at a specific position ω_{l_1} ($l_1 = 1, .., M_1$). In addition, must assign the charge $j = 1$ to the first position of the converter CV_1 ($l_1 = 1$). The constraint (3) demand that the charge $(j+1)$ must be affected on the CV_1 converter where ($l_1 = 1$) at the position ($t_1 = 1$) only if the charge j is to process at any of the positions in $1, \ldots, t_{l_1}$, on one of the $M_1 - 1$ first converters CV_{l_1}. The same way, for two consecutive charges j and $j + 1$, for the last converter CV_{M_1}, constraint (4) has the same meaning. The constraint (5) is set for the start time to process the charges in the converter stage (CV_{l_1}; $l_1 = 1, \ldots, M_1$). Constraints (6)-(7) represent the sequencing in the same charge, exactly for two consecutive operations. The last operation can be started after the first operation reaches its end time and the charge has been brought to the next stage. In addition, constraint (6) (resp. (7)) represents the priority rules between CV_{l_1} and RS_{l_2} (between RS_{l_2} and CC_{l_3} (respectively)). The constraint (8) means that for two consecutive charges to process on the same refining stand (RS_{l_2}), the second charge can only be processed when the first one has reached its end time. The constraint (9) represents the continuity constraints for all the sequences. The constraint (10) defines the inter-sequence correlation setting time between two consecutive sequences to be processed on the same continuous casting machine (CC_{l_3}). Constraint (11) defines the limit of the third stage charging processing time on the machine CC_{l_3}. The constraint (12) means that the stay time (transport) of the charge is finite, and (13) defines the boundary of the set time between the sequence of the third stage. They set the necessary preparation time before the first charging of the sequence after the last charging end time of the previous sequence on the same machine in the third stage.

B. The Objective Function

We define makespan (Sol_{max}) as a function representing the time required to completely process all sequence sets (from the start time of the first charge in the first stage to the termination time of the last charge at the third stage with their inter-sequence dependent setup times.

The objective function we consider for the problem has the following mathematical form:

$$Sol_{max} = \max_{1\le l_3\le M_3}\{\lambda_{S_{l_3}} + z^3_{l_3,S_{l_3}} + pro^3_{l_3,S_{l_3}}\}$$

In order to maximize productivity and the goal is to minimize completion time, we have defined the following goals:

$$Minimize\ Sol_{max} = min \max_{1\le l_3\le M_3}\{\lambda_{S_{l_3}} + z^3_{l_3,S_{l_3}} + pro^3_{l_3,S_{l_3}}\}$$

In this form, the problem is non-linear. Therefore, in order to avoid this situation, we define a new non-negative decision variable z so that we can obtain a new objective (linear) equal to minimizing z and add new constraints:

$$\lambda_{S_{l_3}} + z^3_{l_3,S_{l_3}} + pro^3_{l_3,S_{l_3}} \le z\ \ \forall l_3=1,\ldots,M_3$$

V. The Proposed Contribution to Solve the SCC

Several evolutionary algorithms are developed to solve complex optimization problems, among them we cite [21]. But the real challenge for the SCC problem is to solve instances with an approach intelligent and optimal. Here we develop an adapted neural networks to the SCC with inter-sequence setup times at the last stage using LSTM cells. The adapted neural network trained on a large database of 10 000 use cases solved with CPLEX based on the mathematical model that we presented in the first part of this article.

In the following we detail the evolutionary strategy that we have adopted for this problem but before that we will explain what deep learning is and why exactly the use of the recurrent neural network (RNN) in particular LSTM.

A. Deep Learning

Machine learning is a field of study of artificial intelligence that gives computers the ability to learn from data, that is, to improve their performance at solving tasks without being explicitly programmed. In several areas of machine learning research, it is about creating neural networks.

Deep learning can be defined as special kind of neural networks composed of multiple layers. These networks are better than traditional neural network in persisting the information from previous event. Recurrent neural network is one such machine that has a combination of networks in loop. The networks in loop allow the information to persist. Each network in the loop takes input and information from previous network performs the specified operation and produces output along with passing the information to next network. Some applications require only recent information while others may ask for more from past, exactly the case of the SCC problem. The common recurrent neural networks lag in learning as the gap between required previous information and the point of requirement increases to a large extent. But fortunately Long Short Term Memory Networks [15], a special form of RNN are capable in learning such scenarios.

B. LSTM Neural Networks

Long-term memory (LSTM) is an alternative solution proposed in [15]: the traditional architecture of a Recurrent Neural Network (RNN) that is based on a simple activation function is modified in such a way that the vanishing gradient problem is explicitly avoided, while the learning method remains unchangeable. For more information on this architecture ([4]). But what are the strengths of LSTM ? why the LSTM will be effective to solve the steel continuous casting problem ?

A LSTM neuron network is made up of several cell that have not just one activation function but rather three that are represented as an input gate, a forget gate and an output gate. Each cell remembers the state of the problem treat in several time intervals, and the three gates regulate the flow of information in and out of the cell. The LSTM network is very suitable for classification, processing and prediction based on time series data, because there may be lags of unknown duration between important events in the time series. This is what is needed to schedule tasks between the three stages (CV, RS and CC) of the SCC problem. Also as we explained, LSTM was developed to deal with the explosion and disappearance of gradients that may be encountered when training traditional RNNs. Fig. 2 shows the internal architecture of a LSTM cell.

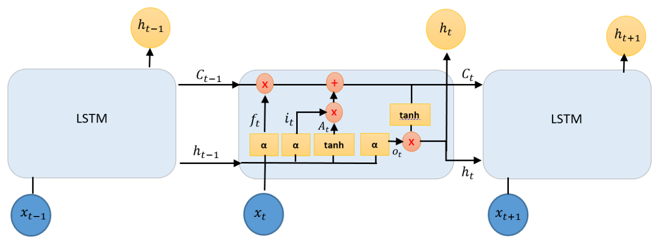

Fig. 2. The Internal Architecture of a LSTM Cell [4].

- **Forget gate:** It is the power of forgetting information. Unlike the classic neural network where it must to memorize all the information in a long sequence, a LSTM has the power to forget unnecessary information that will probably not be used in the prediction. In the LSTM the power to forget non-useful information is represented by the function $f_t = \alpha(W_f * [h_{t-1}, x_t] + b_f)$, where α is a sigmoid function, W_f is the weights and $[h_{t-1}, x_t]$ is the concatenation of the two vectors h_{t-1} and x_t.

- **Input gate:** is responsible for adding relevant information and providing new information. The function that allows this is $C_t = C_{t-1} * f_t + i_t * A_t$ where $i_t = \alpha(W_i * [h_{t-1}, x_t] + b_i$ and $A_t = tanh(W_c * [h_{t-1}, x_t] + b_c)$.

- **Output gate:** This last operation allows to define the current state of the unit. So far, we have forgotten informations, and we have added new informations to the memory. We still need to define the state of the current cell, which represents the output of this cell and which will be the input of the next cell. This is summarized by the following functions: $h_t = \alpha(W_o * [h_{t-1}, x_t] + b_o) * tanh(C_t)$.

C. The Evolutionary Strategy: Training Data, Network Architecture and Choice of Parameters

The proposed neural network will be used as a heuristic to solve the SCC with smart solutions. For this a LSTM is applied to approach the objective function. The proposal to solve the SCC problem is explained in Fig. 3. The system generates a data set in the domain of variables to train a neural network. The objective function of the optimization problem is redefined with the multilayer neural network that transforms the function, allowing to generate a polynomial equation to solve the optimization problem. To define a neural network, it is necessary to establish parameters, such as the training data, the type of neural network, connections, number of layers, activation functions, propagation rules, etc.

Various methods exist to train these networks to produce a specific output for a specific input. Among the current training methods, we have error propagation, which consists of adjusting the network by adapting the weights of each neuron. The use of the partial derivative makes it possible to know in which direction we must modify the weights of our neural network to have the requested output. Also a genetic algorithms are used to train a neural network [19]. By training

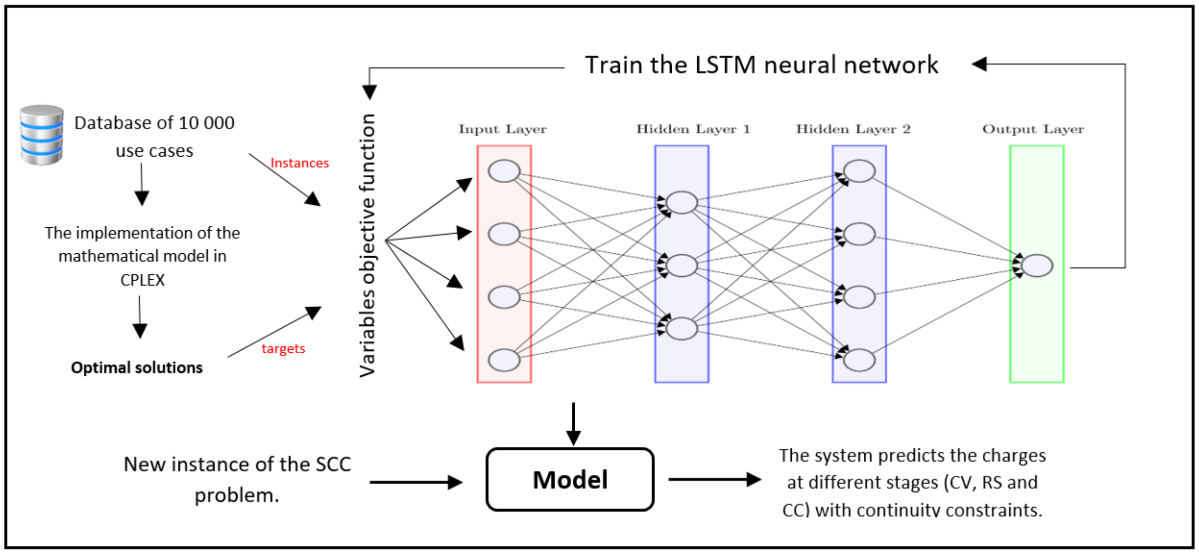
Train the LSTM neural network
Database of 10 000 use cases
Instances
The implementation of the mathematical model in CPLEX
Optimal solutions
targets
Variables objective function
Input Layer
Hidden Layer 1
Hidden Layer 2
Output Layer
New instance of the SCC problem.
Model
The system predicts the charges at different stages (CV, RS and CC) with continuity constraints.

Fig. 3. The Evolutionary Strategy for the SCC.

these networks on a set of data for which the correct output is known, the network will return the appropriate results for similar data. We trained our LSTM on data from 10 000 use cases (instances), where the correct output was generated by the CPLEX based on the mathematical model that we proposed above.

Regarding the architecture, there are several architectures in the literature, but just some differences between them in the neural network language model. The architecture that we propose to solve the steel continuous casting problem was obtained after several tests and it is the one that which gave best results. It can be summarized as follows:

Input sequence represents the input of our LSTM, in our implementation it is encoded by an encoding from 1 to S_{l_3} where S_{l_3} is number of the sequences Seq_{l_3} to process on the machine CC_{l_3}. Note that the sequenes are generated in 128 batches with 5% diversity, where the purpose of adding diversity in the learning stage is to give flexibility to the model and to avoid overfitting. The model predicts the assignment of sequence loads from different stages (CV, RS and CC) with continuity constraints.

The network topology, an architecture with three hidden layers has been applied, each with 256 LSTM units. Each cell LSTM use the "relu" activation function (to have faster convergence compared to other activation functions as tanh or sigmoid). However the output layer, the "softmax" activation function is used to generate the correct normalized probability value in order to select sequential charges in different stages (CV, RS and CC).

As a learning criterion, Adam optimizers is used with a precision of 0,001.

The following algorithm summarizes the main steps :

Algorithm 1 : The proposed LSTM model.

```
- According to the unified law on [0;1], the weights are initialized by random drawing.
- Codage : list all sequence charges, represent each one by a number from 1 to $S_{l_3}$.
- Creating a LSTM network : create four layers, where the three hidden layers have 256 LSTM units for each one. The output layer uses the "softmax" activation function to predict the sequence charges of different stages (CV, RS, and CC) with continuity constraints.
while counter $\leq$ iterMax do
    1. Generate 128 batches with 5% noise.
    2. Train the model.
    3. The learning rate of the Adam optimizer is 0.001.
    4. Update each weight
end while
```

VI. Experimental Tests

All experiments were performed on the Google colab under GPU, namely CPLEX 12.6 was used to determine the optimal solutions for the SCC instances in order to prepare the training data and also to compare with the results of the LSTM.

A. The Learning and Test Rate:

The learning rate gives an idea of the quality of the model. In Fig. 4, we present a graph that represents the learning rate (93%) and the test rate (91%) per epoch.

As can be seen from the convergence of the cost function (Fig. 5), our model performed well and the fact that there is not a large discrepancy between the loss of the training and the loss of the test allows us to conclude that we do not have an overfitting. We have also to mention that the learning rate is 93% and the test rate is 91%.

B. Test Instances

For the instances taken form the literature, we compare our LSTM with the CPLEX solutions, the proposed RNN with LSTM cells has improved the total makespan Sol_{max} and was

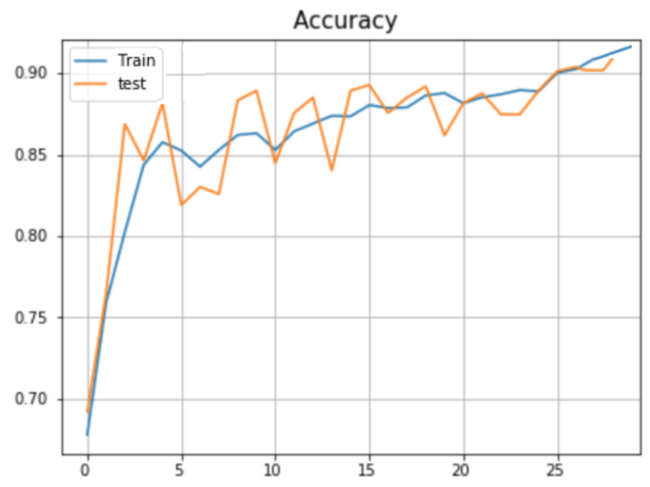

Fig. 4. The Learning and Test Rate per Epoch.

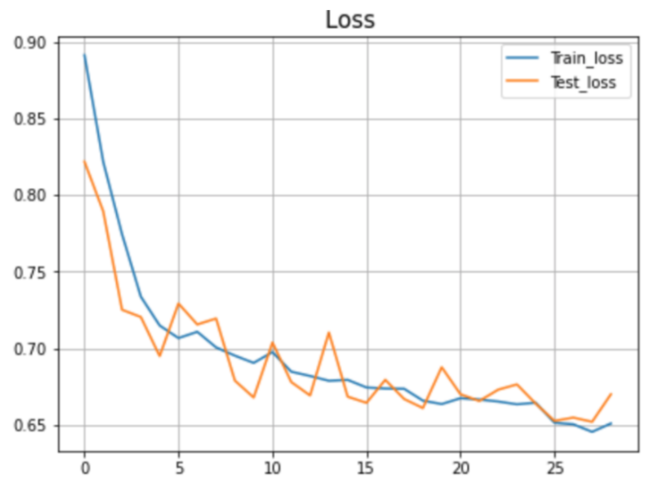

Fig. 5. Convergence of the Cost Function per Epoch.

TABLE I. Results for Different Sequences Sizes with 3 CC and for Several Intervals [$\lambda^{min}_{Seq_{l_3}}, \lambda^{max}_{Seq_{l_3}}$]

Order: [.][.][.][.]	Sol_{max} CPLEX	Time CPU (s)	Sol_{max} LSTM	Time LSTM(s)
[15, 10, 12][5, 5, 5]	1559.00	5.75	1133.23	1.0
[15, 14, 30][5, 5, 5]	1807.01	16.55	1721.10	1.3
[10, 13, 17, 8][5, 5, 5, 5, 5] [18, 11, 23, 15, 27]	1756.72	540.99	1440.67	1.5
[5, 5, 5, 5, 5, 5, 5, 5, 5, 5]	2506,04	2105.31	2420	1.7
[5][5][5]	427.58	1.09	281.39	0.17
[10, 10][10, 10][10, 10] [5, 15, 5, 10][10, 10, 10, 5]	706.07	21.45	651.26	1.1
[5, 5, 5, 5]	1239.34	790.23	1075.37	1.68
[5, 10, 5, 10, 5][10, 10, 10, 5, 5] [5, 5, 5, 5, 5]	1678.23	3367.86	1210.16	1.69
[10, 20, 5, 30][20, 5, 10, 5] [15, 5, 10, 10]	-	-	2910.54	1.57
[20, 10, 5, 40][30, 20, 15][20, 15]	-	-	3372.58	1.63
[35, 15, 25][10, 19, 15, 4, 5] [8, 35, 17]	-	-	3375.44	1.93
[15, 15, 25][20, 14, 25, 11, 5] [15, 35, 17, 10, 30]	-	-	4841.10	2.32

TABLE II. Results on Sequences Sizes Until 10 CC

Number of CC	Total number of charges	Sol_{max} (LSTM)	Time (s)
4	280	4320.28	1.21
4	340	4550.91	2.30
4	470	6330,13	2.42
4	520	6955.39	3.26
4	600	7615.39	3.62
5	235	3872.15	1.12
5	315	4226.28	3.15
5	400	4712.15	3.91
5	625	6980.50	5.76
5	865	10101.45	6.74
6	366	4821.67	3.12
6	515	6316.53	4.75
6	700	7005.22	5.36
8	450	5924.67	3.48
8	635	6473.31	6.85
8	965	8567.92	7.83
10	368	2365.86	2.10
10	823	8391.50	6.34
10	1256	10341.19	8.90

up to 1000 times faster than the CPLEX on CPU as shown in Table I.

Table II shows our tests on randomly generated large instances for which no solution is known. These instances span a large number of charges until 10 CC machines at the last stage. As an example, for the following batches size of a 4 CC system [30,25,20,25][25,30,15,25,15] [30,35,30,20,30 [40,20,55] (number of charges per each sequence) with different [$\lambda^{min}_{Seq_{l_3}}, \lambda^{max}_{Seq_{l_3}}$] ranges of setup times, the CPLEX fails to solve it. However, the LSTM runs it on 1.42 seconds and gave a solution with Sol_{max} = 6330,13. This allows us to say that the obtained numerical results show the efficiency of the proposed recurrent neural network with LSTM cells with a total success of solving all the instances.

VII. Conclusion

In this paper, we have implemented a recurrent neural network with LSTM cells in order to solve the SCC with intersequence dependent setup times and dedicated machines at the last stage known as one of the harder problem in scheduling. Especially, we have shown that with RNN with LSTM cells, one can tackle very large instances arising in complex industrial systems where the number of sequences, of charges or of any devices type (CV, RS or CC) is bigger than 10. Better solutions are obtained with better quality and execution time. The performances of the proposed model are very interesting such that the success rate is 93% and able to resolve large instances while the traditional approaches are limited and fail to resolve very large instances.

One of the future works that we intend to develop is to generalize the approach on a cluster of GPUs in order to deal with more complex and robust cases and to enable solving very large size instances in order to improve the quality of the up to day known solutions. Also, we intend to generalize our approach to similar SCC problems in particular or to solve very complex hybrid flowshops in general. Another feature that could also be envisaged is the lagrangean relaxation for typical hard constraints that we could relax.

ACKNOWLEDGMENT

The author would like to thank the Editor-in-chief and anonymous reviewers for their comments and suggestions to improve the quality of the paper

REFERENCES

[1] Aqil S. and Allali K. Two efficient nature inspired meta-heuristics solving blocking hybrid flow shop manufacturing problem. *Engineering Applications of Artificial Intelligence*, *100*, 104196, (2021).

[2] Atighehchian A., Bijari M. and Tarkesh H. A novel hybrid algorithm for scheduling steel-making continuous casting production. *Computers and Operations Research*, *36(8)*: 2450-2461, (2009).

[3] Basu, S. and Dutta, G. A Survey of the Non-Optimization techniques used in an integrated steel plant. *Management Dynamics*, *6*: 33–68, (2006).

[4] Berrajaa, A. and Ettifouri, E. H. The Recurrent Neural Network for Program Synthesis. *In International Conference on Digital Technologies and Applications*, 77-86, (2021).

[5] Blazewicz J., Ecker K. H., Pesch E., Schmidt G. and Weglarz J. Scheduling computer and manufacturing processes. *springer science and Business media*, (2013).

[6] Cappel J., Kaiser H. P. and Schlüter J. Time management at the HKM-Huckingen BOF-Shop. *In Proceedings 4 th European Oxygen Steelmaking Conference (EOSC)*, *Vol.12*: 15.5, (2003).

[7] Chakraborti N., Kumar R. and Jain D. A study of the continuous casting mold using a pareto-converging genetic algorithm. *Applied Mathematical Modelling*, *25(4)*: 287-297, (2001).

[8] Chang, P. C., and Chen, S. H. Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times. *Applied Soft Computing*, *11(1)*: 1263-1274, (2011).

[9] Chen Z. Control of constrained moving-boundary process with application to steel continuous casting. *(Doctoral dissertation, University of Illinois at Urbana-Champaign)*, *36(8)*:2450-2461, (2020).

[10] Ferretti I., Zanoni S. and Zavanella L. Production–inventory scheduling using Ant System metaheuristic. *International Journal of Production Economics*, *104(2)*: 317-326, (2008).

[11] Foumani S. N. A., Guo C. and Luk W. An Analysis of Alternating Direction Method of Multipliers for Feed-forward Neural Networks. *arXiv preprint arXiv:2009 :02825*, (2020).

[12] Han D., Tang Q., Zhang Z., Yuan L., Rakovitis N., Li D. and Li J. An Efficient Augmented Lagrange Multiplier Method for Steelmaking and Continuous Casting Production Scheduling. *Chemical Engineering Research and Design*, *168*: 169-192, (2021).

[13] Hao J., Liu M., Jiang S. and Wu C. A soft-decision based two-layered scheduling approach for uncertain steelmaking-continuous casting process. *European Journal of Operational Research*, *244(3)*: 966-979, (2015).

[14] Helal, M., Rabadi, G. and Al-Salem, A. A tabu search algorithm to minimize the makespan for the unrelated parallel machines scheduling problem with setup times. *International Journal of Operations Research*, *3*: 182–192, (2006).

[15] Hochreiter S. Long Short-Term Memory. *Neural Computation*, *9(8)*, (1997).

[16] Hohn W., Konig F. G., Mohring R. H. and Lubbecke M. E. Integrated sequencing and scheduling in coil coating. *Management Science*, *57(4)*: 647-666, (2011).

[17] Ko C. H. and Wang S. F. Precast production scheduling using multi-objective genetic algorithms. *Expert Systems with Applications*, *38(7)*: 8293-8302, (2011).

[18] Kumar V., Kumar S., Tiwari M. K. and Chan F. T. S. Auction-based approach to resolve the scheduling problem in the steel making process. *International journal of production research*, *44(8)*: 1503-1522, (2006).

[19] Lamos-Sweeney, J. D. Deep learning using genetic algorithms. *Rochester Institute of Technology*, (2012).

[20] Lee H. S., Murthy S. S., Haider S. W. and Morse D. V. Primary production scheduling at steelmaking industries. *IBM Journal of Research and Development*, *40(2)*: 231-252, (1996).

[21] Lee T. and Loong Y. A review of scheduling problem and resolution methods in flexible flow shop. *International Journal of Industrial Engineering Computations*, *10(1)*: 67-88, (2019).

[22] Michael L. P. Scheduling: theory, algorithms, and systems. *Springer*, (2018).

[23] Naphade K. S., Wu S. D., Storer R. H. and Doshi B. J. Melt scheduling to trade off material waste and shipping performance. *Operations Research*, *49(5)*: 629-645, (2001).

[24] Pan Q. K., Wang L., Mao K., Zhao J. H. and Zhang M. An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process. *IEEE Transactions on Automation Science and Engineering*, *10(2)*: 307-322, (2012).

[25] Pan Q. K. An effective co-evolutionary artificial bee colony algorithm for steelmaking-continuous casting scheduling. *European Journal of Operational Research*, *250(3)*: 702-714, (2016).

[26] Peng K., Pa, Q. and Zhang, B. An improved artificial bee colony algorithm for steelmaking–refining–continuous casting scheduling problem. *IEEE Chinese journal of chemical engineering*, *26(8)*: 1727-1735, (2018).

[27] Tan Y., Zhou M., Zhang Y., Guo X., Qi L. and Wang Y. Hybrid scatter search algorithm for optimal and energy-efficient steelmaking-continuous casting. *IEEE Transactions on Automation Science and Engineering*, *17(4)*: 1814-1828, (2020).

[28] Tang L., Liu J., Rong A. and Yang Z. A review of planning and scheduling systems and methods for integrated steel production. *European Journal of Operational Research*, *133(1)*: 1-20, (2001).

[29] Tang L. and Liu G. A mathematical programming model and solution for scheduling production orders in Shanghai Baoshan Iron and Steel Complex. *European Journal of Operational Research*, *182(3)*: 1453-1468, (2007).

[30] Tang L., and Wang G. Decision support system for the batching problems of steelmaking and continuous-casting production. *Omega*, *36(6)*: 976-991, (2008).

[31] Wu X., Jin H., Ye X., Wang J., Lei Z., Liu Y., ... and Guo Y. Multiscale Convolutional and Recurrent Neural Network for Quality Prediction of Continuous Casting Slabs. *Processes*, *9(1)*: 33, (2021).

[32] Xu Z., Zheng Z., and Gao X. Energy-efficient steelmaking-continuous casting scheduling problem with temperature constraints and its solution using a multi-objective hybrid genetic algorithm with local search. *Applied Soft Computing*, *95*: 106554, (2020).

[33] Yang J. M., Che H. J., Dou F. P. and Zhou T. Genetic algorithm-based optimization used in rolling schedule. *Journal of Iron and Steel Research International*, *15(2)*: 18-22, (2008).