Advances in Industrial Engineering, Autumn 2019, 53(4): 127-147
DOI: 10.22059/jieng.2021.306226.1732

RESEARCH PAPER

Production Scheduling Optimization Algorithm for the Steel-Making Continuous Casting Processes

Mahdi Nakhaeinejad*

Department of Industrial Engineering, Yazd University, Yazd, Iran

Received: 14 July 2020, Revised: 13 January 2021, Accepted: 02 February 2021

Abstract

This paper investigates the steel-making continuous casting (SCC) scheduling problem. SCC is a high temperature and large-scale process with batch production at the last stage that was identified as the key process of modern iron and steel enterprises. This paper presents a mathematical model for scheduling the SCC process. The model is developed as a Mixed Zero-One Linear programming (MZOLP) according to actual situations of SCC. The objective is scheduling a set of charges (jobs) to minimize the earliness and tardiness penalty costs as well as the charge waiting time cost. The solution methodology is developed based on a Branch and Bound (B&B) algorithm. A heuristic method presented at the beginning of the search to compute an initial upper bound. A lower bound and an upper bound are developed and a method for reducing branches is established based on the batch production in the continuous casting (CC) stage. Moreover, branching schemes are proposed. The B&B algorithm presenting the initial upper bound, the lower and upper bound, the method for reducing branches, and branching schemes is tested on a set of instances. The analysis shows the efficiency of the proposed features of the algorithm.

Keywords:
Steelmaking;
Continuous Casting;
Production Scheduling;
Branch and Bound Algorithm

Introduction

Production scheduling, one of the most challenging problems in the steel industry, is a key component for productivity in manufacturing. It has been distinguished as one of the most challengeable and complicated problems for manufacturers.

The steelmaking-Continuous Casting (SCC) process plays an important role in modern steel production process. This process is a high-temperature complicated process that runs continuously to convert products from liquid (molten steel) into solid (bloom). Also, the SCC process is the bottleneck in steel production [1]. Therefore, effective scheduling of the SCC process is vital.

Scheduling in the SCC process is to specify the sequence, time and machine on which the molten steel should be processed at stages from electronic arc furnace to continuous casting [2, 3]. Cast sequencing which defined at the lot planning level determines the cast sequence and charge sequences for each cast. Then, SCC scheduling decides the starting time and also determine the machine for each charge by considering constraints and technological requirements to insure the practical feasibility of the resulting [4]. SCC process involves four stages, that in each stage there are parallel machines, as shown in Fig. 1.

* Corresponding author: (M. Nakhaeinejad)
Email: m.nakhaeinejad@yazd.ac.ir

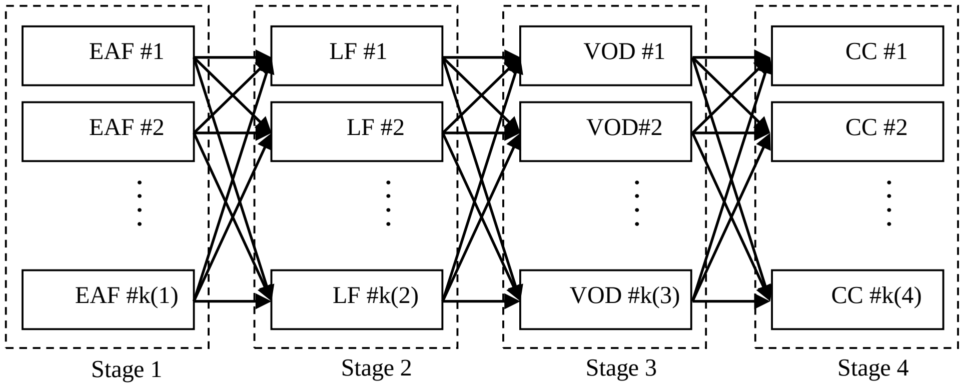

Fig. 1. The steel making- continuous casting process

In the first stage, the molten steel is combined with scrap through an Electronic Arc Furnace (EAF) and the molten steel impurities are reduced to desired levels by burning with oxygen. In the second stage, the molten steel with the main alloy elements from the previous stage is poured into ladles that are carried by a crane to a refining furnace or ladle furnace (LF) for refining. In this stage, the chemical mixture of the steel adjusts to produce specific steel grades. The grade is one of the quality characteristics of products that determines the chemical and physical properties of the charge. The basic unit of SCC production, which denoted as a "job" in SCC scheduling, is called charge. The third stage is vacuum degassing (VD) and vacuum oxygen decarburization (VOD). The VD process is conducted at high pressure. Under this condition, much of the dissolved hydrogen and carbon monoxide gases in the liquid metal are evacuated. Also, in the VOD process, the injection of pure oxygen into the molten steel is used to burn out dissolved carbon, which is evacuated away. This stage is only used for specific high-quality steel products. Finally, in the last stage of continues casting (CC), the graded liquid steel from the previous stage is poured into the input unit of a continuous caster called Tundish and is solidified into blooms.

Although the production structure of the above SCC process has the general specifications of the hybrid flow shop (HFS) problem, the SCC scheduling problem is different from the classic HFS. SCC scheduling needs special requirements of steel production leading to be different from HFS [5]. There are strict requirements to provide continuity of the production process including processing time on various machines and transportation and waiting time between operations [2]. A scheduling model for SCC must consider equipment availability, the temperature of each charge at each stage and chemistry constraints, between subsequent charges and the batch production at the continuous caster stage [6]. These requirements make the SCC problem more complicated than HFS. Gupta et al. [7] showed scheduling two-stage HFS with the objective function of minimizing the makespan is NP-complete. So, the SCC scheduling problem is NP-hard too.

Several techniques for SCC scheduling of iron and steel production have been proposed by other researchers. Review of previous research on integrated steel production planning and scheduling investigated by Tang et al. [1]. Tang et al. [2] developed a non-linear model for scheduling SCC and converted it into a linear programming model for easy solution. Harjunkoski and Grossman [6] studied a decomposition algorithm for the scheduling of a steel plant production using mathematical programming model. Their algorithm is based on disaggregating the original problem. Tang et al. [4] constructed an integer programming formulation with a separable structure and provided a solution by combining the Lagrangian relaxation, dynamic programming, and heuristics. Xuan and Tang [5] proposed an integer-programming model for hybrid flowshop with batch production at the last stage for iron and steel industry. They studied a batch decoupling based Lagrangian relaxation algorithm for this problem. Atighehchian et al. [3] proposed an iterative algorithm for SCC scheduling. Their algorithm, named HANO, is according to the combination of ant colony optimization and non-linear optimization method. Missbauer et al. [8] describe the models, algorithms and implementation results of a computerized scheduling system for the steelmaking-continuous casting process of a steel plant in Austria. Sun et al. [9] introduced a surrogate subgradient algorithm for Lagrangian relaxation to solve the problem of SCC scheduling. Dao-Iei et al. [10] proposed an optimization model to improve the efficiency and performance of production planning in the SCC process. The optimization model has combined with parallel-backward inferring algorithm and genetic algorithm. They also presented a simulation model based on cellular automata to analyze and evaluate the production plans. Witt and Voss [11] presented a case study for the production planning of a German steel producer and included the lot-sizing problem to provide the restrictions for the mid- to long-term planning of steel producers. Wei and Liang-liang [12] considered a batch split scheduling to solve continuous and ingot casting schedule of mixed whole/half charging plan for steel production. They solved the models by a heuristic algorithm. Gui-rong et al. [13] proposed the cross-entropy method to optimize the SCC production scheduling problem lead to increase productivity and reduce energy consumption. They show the entropy method is effective in this problem with complicated technological routes. Touil et al. [14] addressed a simulated annealing algorithm restricted by a tabu list to maximize the production and minimize the processing time in the SCC by optimizing the sequence of the group of jobs with the same chemical specifications. Hadera et al. [15] proposed a MILP formulation to develop a continuous-time model with energy-awareness to optimize the daily production schedules and the electricity purchase including the load commitment problem. They proposed a bi-level heuristic algorithm to deal with industrial cases. Nastasi et al. [16] presented a novel route planning system for steel coils. They take into account customers' quality requirements with their newly proposed approach. Armellini et al. [17] proposed a general model for planning and scheduling steelmaking and casting activities. They developed a simulated annealing approach with a solution space created by job permutations, which uses as a submodule chronological constructive procedure that assigns processing times and resources to jobs. Li et al. [18] proposed a new non-dominated sorting genetic algorithm with elite strategy (NSGA2) according to the production scheduling method for the complex SCC production process which is formed of multiple refining ways. Peng et al. [19] studied a hybrid flowshop rescheduling problem in the SCC process with disruption and controllable processing times in the last stage of SCC process. They developed an improved Artificial Bee Colony algorithm to solve the problem. Jiang et al. [20] focused on the uncertain scheduling problem of the SCC manufacturing system and proposed a multi-objective soft scheduling. They introduced a soft-form schedule including critical decisions and characteristic indicators against random disturbances. Fazel Zarandi and Dorry [21] developed a fuzzy mixed integer linear programming model for the SCC scheduling problem. They developed a hybrid algorithm to solve the SCC scheduling problem. Their algorithm was based on a particle swarm optimization and fuzzy linear programming methods. Rahal et al. [22] presented a robust optimization method for SCC scheduling under uncertainty. They compared their method to the deterministic scheduling method. Their adaptive robust hybrid scheduling presented an attractive trade-off between solution conservatism and excessive scheduling modifications. Cui et al. [23] investigated a hybrid flowshop scheduling (HFS) problem in the SCC process. They transformed the SCC scheduling problem into a DC (difference of convex functions) programming problem by relaxing the machine capacity constraint. They proposed an effective and efficient deflected surrogate subgradient method with global convergence to solve the Lagrangian dual (LD) problem. Long et al. [24] studied a dynamic scheduling problem considering the uncertainty of the job release time in SCC production processes. They proposed a novel robust dynamic scheduling approach based on release time series forecasting. They formulated an evolutionary algorithm by combining a genetic algorithm with a local search strategy.

It may also be seen in the literature that many of the researches have focused on Branch and Bound (B&B) as a strategy for solving some scheduling problems. Portman et al. [25] proposed an improvement of lower bound and a new method of hybrid flow shop scheduling. Their method was based on a Genetic Algorithm in a B&B algorithm. Akkan and Karabat [26] presented a B&B algorithm for the two-machine flow shop scheduling with a new lower bounding scheme based on the network formulation. Moursli and Pochet [27] studied a B&B algorithm using a new branching scheme for hybrid flow shop scheduling to minimize makespan. They also considered several upper and lower bounding schemes. Haouari et al. [28] proposed an exact B&B algorithm for the two-stage hybrid flowshop problem with at least two identical machines in each stage incorporating several characteristics such as presenting a schedule as a permutation of jobs, fast lower bounds, effective heuristics, adjustment procedures, and dominance rules. Ng et al. [29] considered a B&B algorithm for two-machine flow shop problem with deteriorating jobs. They proposed several dominance properties, some lower bounds, and an initial upper bound by using a heuristic algorithm and applied to the elimination process of a B&B algorithm. Ranjbar et al. [30] developed two B&B algorithms for the scheduling of jobs on a specified number of identical parallel machines when the processing time is stochastic. Khoudi and Berrichi [31] studied scheduling and preventive maintenance (PM) on a single machine problem. They presented jobs sequencing and PM planning to minimize total tardiness and machine unavailability. They proposed a bi-objective exact algorithm based on the bi-objective B&B method to find the efficient set. Tanaka et al. [32] proposed a B&B algorithm with new components for solving large pre-marshalling problems. They developed existing lower bounds and proposed two new lower bounds by relaxation of the pre-marshalling problem to give tight bounds in specific situations. They introduced generalized dominance rules that cause search space reduction, and a memorization heuristic that leads to feasible solutions quickly. Bunel et al. [33] exploited the Mixed Integer Linear Programming (MIP) formulation for Piecewise Linear Neural Network to propose algorithms based on Branch-and-Bound. They identified methods that combine the strengths of multiple existing approaches, accomplishing significant performance improvements. They also introduced an effective branching strategy that allows for high input dimensional problems with convolutional network architecture.

This paper developed a new model, considering all technological constraints needed in the practical and real world, simultaneously such as 1) no conflict between consecutive charges processed on the same machine, 2) no conflict between consecutive operations for the same charge, 3) waiting time of charges between the processing at different stages, 4) total waiting time of each charge in the system, 5) batch production in the continuous caster stage. Different methods are often considered by researchers for SCC scheduling, while, in this paper with a newly proposed approach, heuristic-based on B&B are taken into account. In fact, the novelty of this paper is proposing a B&B algorithm for SCC scheduling. Although the B&B algorithm have been presented for HFS scheduling in the literature, but for SCC scheduling with their own features have not been considered in recent researches. Therefore, this paper presents an effective B&B algorithm for SCC scheduling. The main feature of the proposed B&B algorithm is 1) initial upper bound, 2) upper bounding scheme based on scattering charges on the machines in each stage, 3) branching strategies, and 4) reducing branches according to the batch production in the last stage.

The reminder of this paper is organized as follows. Section 2 presents a mixed zero-one nonlinear programming (MZONLP) model for the SCC scheduling. In section 3, the MZONLP model is transformed into a MZOLP model. Section 4 describes a B&B solution methodology incorporating upper and lower bounds, branching schemes and a method for reducing branches.

Computational results using the collected practical data are reported in section 5 to evaluate the performance of the proposed model and algorithm. Finally, section 6 concludes this study.

SCC scheduling model

In this section, the SCC scheduling problem is formulated. The problem consists of scheduling I charges at four stages of EAF, LF, VD & VOD, and CC. At each stage, any charge can be performed on one of $k_{(j)}$ parallel identical machines, where j is the index for stage. At the last stage (CC), Charges are processed as G batches. All the charges from the same batch must be processed on one machine of the CC stage consecutively while satisfying given precedence constraints for the charges in each batch. The precedence constraints are defined by lot planning and is given as one of the inputs data for the SCC scheduling model. The SCC scheduling should determine when and where (on which machine) each charge at each production stage should be processed. The assumptions considered for the SCC scheduling model are as follows:

1. All charges follow the same production process route including 4 stages: EAF, LF, VD & VOD, and CC. A charge can be processed on any of the identical parallel machines at each stage.
2. Each machine in each stage processes only one charge at a time.
3. Each charge is processed only on one machine at a time.
4. The process of charges is non-pre-emptive.
5. Transfer times between stages should be considered based on the fact that charges are molten steel at high temperature.
6. All the charges from the same batch (a cast) should be processed on the same caster.
7. The charges within a cast are processed consecutively. In fact, cast break is not allowed in the caster stage.
8. At the casting stage, precedence constraints of the charges within a batch should be met.
9. Setup time is required for changing cast on a continuous caster machine.
10. Since the waiting time of charges between stages causes temperature drop, this drop effect is considered in the model.
11. According to the fact that production planning in the steel industry is a make- to- order system, both earliness and lateness cost of each charge should be considered, e.g. for inventory cost or compensation cost to customers.

The SCC scheduling model requirements are:

1. The objective function is continuity of the production process as well as on-time delivery of final products through minimizing a cost function consisting of charge waiting time from operation to operation and earliness/ lateness of blooms at the end of the continuous caster.
2. The constraints considered for this model are to guarantee no conflict between consecutive charges processed on the same machine and also no conflict between consecutive operations for the same charge. Charges waiting time between different stages and total waiting time of each charge in the system are other important constraints that should be considered in the model. Also, the constraints should result in a batch production in the continuous caster stage.
3. The decisions that the proposed model should be made is on which machine each charge should be processed at each production stage. Also, the start time of processing charges on the machines is another decision-making factor.

Parameters and Variables

The SCC scheduling model could be given by following notations.

i index of charges, $i=\{1,\ldots, I\}$, where I is the number of production charges,

j index of stages, where j=1, 2, 3, and 4 are referred to as EAF, LF, VD and VOD, and CC, respectively,

$K_{(j)}$ number of machines in the jth stage,

g index of cast, $g=\{1,\ldots, G\}$, where G is the total number of casts

p_g the total number of charges in cast g,

$f(g,p)$ the pth charge in the cast g,

d_i due date of charge i,

$C1_{ij}$ cost of waiting time for charge i after being finished at stage j,

$C2_i$ coefficient of penalty cost in each time unit for charge i when completed before its due date (earliness cost),

$C3_i$ coefficient of penalty cost in each time unit for charge i when completed after its due date (lateness cost),

T_{ijk} processing time of charge i on kth machine of stage j,

t_j transportation time from machine j to machine ($j+1$), where j is equal to 1, 2, and 3,

s_{ij} setup time of charge i on a machine at stage j. s_{ij} for all i, j is equal to zero except when j=4 and i is the first charge in a cast,

mw maximum waiting time allowed before casting stage,

mtw maximum waiting time allowed for each charge in the system from EAF to CC,

M is a very large number.

Decision variables:

st_{ijk} starting time of charge i on kth machine at stage j

$$x_{ijk} = \begin{cases} 1 & \text{if charge i is processed on kth machine of stage j} \\ 0 & \text{otherwise;} \end{cases}$$

Model Formulation

The scheduling problem is scheduling I charges at four stages, which are processed as G batches at the CC stage. At each stage, the job can be performed on any of $k_{(j)}$ parallel identical machines. The proposed model is different from the ones available in the literature. This paper maintains that the proposed model is not in the literature and that it has real applications in industry. Using the above parameters and variables, the SCC scheduling problem formulation is as follows:

$$\begin{aligned}\min z = &\sum_{i=1}^{I}\sum_{j=1}^{3} C1_{ij}\Big(\sum_{k=1}^{k(j+1)} st_{i,j+1,k} - \sum_{k=1}^{k(j)} st_{i,j,k} - \sum_{k=1}^{k(j)} (T_{i,j,k} * x_{i,j,k}) - \mathrm{t}_j\Big) + \\ &\sum_{i=1}^{I} C2_i \max(0, d_i - \sum_{k=1}^{k(4)} (st_{i,4,k} + x_{i,4,k} * \mathrm{T}_{i,4,k}) + \\ &\sum_{i=1}^{I} C3_i \max(0, \sum_{k=1}^{k(4)} (st_{i,4,k} + x_{i,4,k} * \mathrm{T}_{i,4,k}) - d_i)\end{aligned} \quad (1)$$

Subject to

$st_{i,j,k} * x_{i',j,k} \geq st_{i',j,k} * x_{i,j,k} + T_{i',j,k} * x_{i',j,k} + s_{i,j} * x_{i,j,k} * x_{i',j,k}$ for i=1,…,I; $i' < i$; j=1,…,4; k=1,…,$k(j)$. (2)

$$\sum_{k=1}^{k(j+1)} st_{i,j+1,k} - \sum_{k=1}^{k(j)} st_{i,j,k} \geq \sum_{k=1}^{k(j)} (T_{i,j,k} * x_{i,j,k}) + t_j; \quad \text{for } i=1,\dots,I;\ j=1,2,3. \tag{3}$$

$$\sum_{k=1}^{k(4)} st_{i,4,k} - \sum_{k=1}^{k(3)} st_{i,3,k} - \sum_{k=1}^{k(3)} (T_{i,3,k} * x_{i,3,k}) - t_3 \leq mw; \quad \text{for } i=1,\dots,I. \tag{4}$$

$$\sum_{j=1}^{3} \left(\sum_{k=1}^{k(j+1)} st_{i,j+1,k} - \sum_{k=1}^{k(j)} st_{i,j,k} - \sum_{k=1}^{k(j)} (T_{i,j,k} * x_{i,j,k}) - t_j \right) \leq mtw; \quad \text{for } i=1,\dots,I; \tag{5}$$

$$\sum_{k=1}^{k(j)} x_{i,j,k} = 1; \quad \text{for } i=1,\dots,I;\ j=1,\dots,4. \tag{6}$$

$$st_{i,j,k} \leq x_{i,j,k} * M; \quad \text{for } i=1,\dots,I;\ j=1,\dots,4;\ k=1,\dots,k(j). \tag{7}$$

$$st_{f(g,p),4,k} - st_{f(g,p-1),4,k} - (x_{f(g,p-1),4,k} * T_{f(g,p-1),4,k}) = 0; \quad \text{for } g=1,\dots,G;\ p=2,\dots,P_g;\ k=1,\dots,k(4). \tag{8}$$

$$x_{i,j,k} = 0 \text{ or } 1; \quad \text{for } i=1,\dots,I;\ j=1,\dots,4;\ k=1,\dots,k(j). \tag{9}$$

$$st_{i,j,k} \geq 0; \quad \text{for } i=1,\dots,I;\ j=1,\dots,4;\ k=1,\dots,k(j). \tag{10}$$

The objective function (1) is scheduling charges to minimize a cost function that composed of three terms: penalty cost of charges waiting time, earliness penalty cost, and lateness penalty cost, respectively. Constraints (2) guarantee that no overlap exists between two contiguous charges on the same machine. In other words, only when the preceding charge is finished, the immediately next one can be started. Constraints (3) ensure operation precedence among the stages for a charge. In fact, for contiguous operations on the same charge, only when the preceding operation is finished and the charge is transported to the next machine, the next operation on the charge can be started. By these constraints, a charge will not be at more than one stage at the same time. Constraints (4) set an upper limit for the waiting time before the casting stage. According to the specifications of casting stage, considering these constraints is vital. Also, there is a maximum total waiting time allowed for a charge in the system which is shown by constraints (5). In fact, the charge could not spend more than a certain time, wait time, in the system because the charge temperatures must be within a required range. Considering constraints (4) and (5) lead to don’t fail the grade needed in production and the products achieve its quality specification without further cost. Constraints (6) require that a charge at each stage can be assigned to only one machine on that stage. Constraints (7) ensure that charge starting time on machine at any stage is defined when it is assigned to that machine; otherwise, this variable (st_{ijk}) must be zero because the charge didn't assign to the machine on that stage and x_{ijk}= 0. Constraints (8) impose batch production in the continuous casting stage. The role of these constraints is to determine the starting time of each charge and sequence of all casts that processed in the same caster in a way that there is no time between two adjacent charges on a cast. Finally, constraints (9) and (10) determine the value range of the variables.

Model linearization

The proposed model in the previous section is a MZONLP model. Since the model is nonlinear, model linearization will help to solve the model easier and faster. It is clear that, the non-linearity of the model is related to two terms. The first one is because of penalties of earliness and lateness in objective function and the second one is related to the constraints (2).

To deal with the non-linearity in objective function, two sets of new variables should be defined as below:

$$q2_i = \max(0, d_i - \sum_{k=1}^{k(4)} (st_{i,4,k} + x_{i,4,k} * T_{i,4,k})). \tag{11}$$

$$q3_i = \max(0, \sum_{k=1}^{k(4)} (st_{i,4,k} + x_{i,4,k} * T_{i,4,k}) - d_i). \tag{12}$$

It is clear that $q2_i$ and $q3_i$ are nonnegative, and

$$d_i - \sum_{k=1}^{k(4)} (st_{i,4,k} + x_{i,4,k} * T_{i,4,k}) \leq q2_i. \tag{13}$$

$$\sum_{k=1}^{k(4)} (st_{i,4,k} + x_{i,4,k} * T_{i,4,k}) - d_i \leq q3_i. \tag{14}$$

So, by replacing constraint (1) by constraint (15) as follows the linearization takes place for constraint (1):

$$\min z = \sum_{i=1}^{I} \sum_{j=1}^{3} C1_{ij} (\sum_{k=1}^{k(j+1)} st_{i,j+1,k} - \sum_{k=1}^{k(j)} st_{i,j,k} - \sum_{k=1}^{k(j)} (T_{i,j,k} * x_{i,j,k}) - \mathrm{t}_j) + \sum_{i=1}^{I} (C2_i * q2_i) + \sum_{i=1}^{I} (C3_i * q3_i) \tag{15}$$

To deal with non-linearity in constraints (2), substituting them with following form (Eq. 16) is relevant.

$$st_{i,j,k} \geq st_{i',j,k} + T_{i',j,k} * x_{i',j,k} + s_{i,j,k} * x_{i,j,k} - M(1 - x_{i',j,k} + 1 - x_{i,j,k}); \quad \text{for} \quad i=1,2,\ldots,I; \quad i' < i; \quad j=1,2,\ldots,J; \; k=1,2,\ldots,k(j). \tag{16}$$

After this treatment, substituting constraints (1) and (2) by constraints (15) and (16), respectively, the model changes to the MZOLP form, where has been much simplified to be solved. The MZOLP model results in much less computation time than the previous model (MZONLP). For instance, in a problem where there are two machines at all stages that should process 10 charges which are grouped into 3 casts, the computational time on a personal computer with a Pentium (R) Dual-Core- 2.60 GHz CPU and 2 GB memory is about a few minutes with MZOLP while it takes over 7 hours to be solved with MZONLP. Fig. 2 shows the result of the SCC schedule for this example by Gantt graph where the horizontal axis shows time in minutes, each horizontal line is a machine and the length of a pane is the processing time.

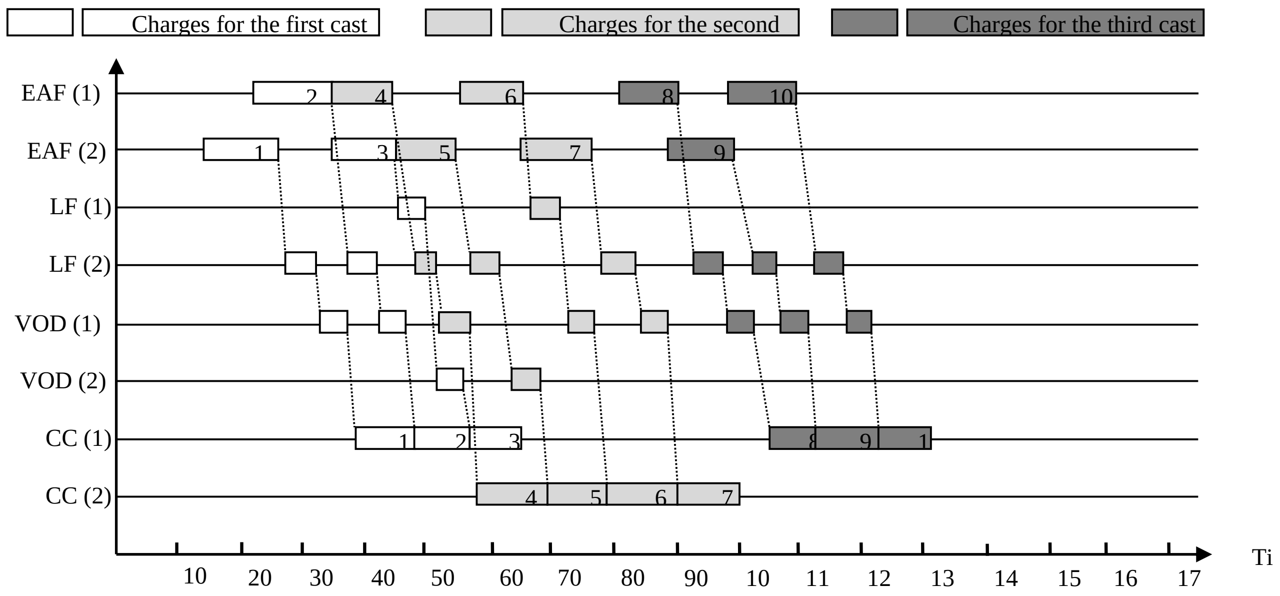

Fig. 2. The SCC schedule results for the problem solved by MZOLP model

As the Gantt chart in Fig. 2 shows the performance of the proposed SCC scheduling model is appropriate for applying in practical. All constraints needed in the real world now depicted in Fig. 2 for the presented example. In the continuous casting stage, there is no break cast in schedule charges of each cast. There is no conflict between consecutive charges processed on the same machine and also no conflict between consecutive operations for the same charge. Charges waiting time between different stages and total waiting time of each charge in the system are other constraints that observed.

The proposed B&B algorithm

This section gives details of a B&B algorithm for the proposed the SCC scheduling model. B&B algorithm is designed for SCC scheduling problem based on the features of this problem. The most important feature of this problem is its decision making, x_{ijk} that its indexes means i, j, and k that is related to charges, stages, and machines could be simply considered for branching in the algorithm. This section explains how the proposed heuristic method generates an initial upper bound, describe branching rules, propose lower and upper bound, and derive a method to reduce the branches in the search tree. This paper proposed the B&B algorithm, which is basically according to the algorithm proposed by [34]. The following symbols are employed throughout this algorithm:

$S_0=\{ ijk | x_{ijk}$ is fixed to 0}

$S_1=\{ ijk | x_{ijk}$ is fixed to 1}

$S_*=\{ ijk | x_{ijk}$ is still undecided}

P is the MZOLP model of SCC scheduling problem

$LP(S_0, S_1, S_*)$ is the linear relaxed problem of P denoted by Eq. 17.

$$\begin{aligned} \min \quad & f(st_{i,j,k}, x_{i,j,k}) \\ & (x, st) \in C \\ & st \geq 0 \\ & x_{i,j,k} = 0; \quad (ijk \in S_0) \\ & x_{i,j,k} = 1; \quad (ijk \in S_1) \\ & 0 \leq x_{ijk} \leq 1; \quad (ijk \in S_*) \end{aligned} \tag{17}$$

F^* is the currently best objective value found during the calculation (upper bound).

The B&B algorithm is implemented, which each node corresponds to a partial decision $LP(S_0, S_1, S_*)$.

Initial upper bound

Initial upper bound means calculations a feasible solution to problem *P* at the root of the search tree in the B&B algorithm. So, the related objective value could be used as initial F^*. Applying an effective initial upper bound helps to restrict the search and enables more solutions to be deleted at an early stage. In fact, the smaller the initial F^*-value, the sharper the bounding test will be.

The heuristic method for the initial upper bound in the proposed algorithm is based upon scattering the charges on the machines available at stages 1, 2, and 3 and scattering the casts on the machines available at stage 4 (CC stage). I1 and I2 are the algorithms that described this heuristic method. Variables of stages 1, 2, and 3 are produced by algorithm I1 and variables related to the CC stage are determined by algorithm I2.

Algorithm I1:

Step 1. Assign the charges to the machines, based on the charges sequence, starting from the first machine and going sequentially forward until the last machine in that stage has been reached.

Step 2. Delete the charges that are assigned, from further consideration.

Then, do the following steps repeatedly, until all charges are assigned:

Step 3. Let C_k for k=1,…, $k_{(j)}$ be the sum of processing times of the charges that have already been scheduled on machine *k*.

Step 4. Choose the machine with the smallest C_k at the stage and assign the next charge.

Step 5. Delete the assigned charge, from further consideration.

For the CC stage, the assignment is for casts instead of charges. So, the algorithm is modified as follows:

Algorithm I2:

Step 1. Assign the casts to the machines, based on the casts' sequence, starting from the first machine and going sequentially forward until the last machine in the CC stage has been reached.

Step 2. Delete the casts that are assigned, from further consideration.

Then, do the following steps repeatedly, until all casts are assigned:

Step 3. Let C_k for k=1,…, $k_{(4)}$ be the sum of charges processing times that have already been scheduled on machine *k*.

Step 4. Choose the machine with the smallest C_k at the CC stage and assign the next cast.

Step 5. Delete the assigned cast, from further consideration.

Based on algorithm I2, there is no break cast in scheduled charges. The process precedence is also satisfied by algorithm I1 and I2. It is clear that the other constraints of problem *P* are satisfied by these two algorithms. So, the initial solution (initial upper bound) obtained by the above algorithms is feasible and could be used as an initial upper bound to restrict the size of the search tree.

Lower bounding and upper bounding strategy

This section describes the lower and upper bounds that should be computed at each node of the B&B algorithm. Lower and upper bounding strategies are required for accelerating the search of the optimal solution.

Lower bounding strategy

The lower bound computations of the proposed B&B algorithm are based on a linear programming (LP) relaxation. Solving the LP relaxation of the problem *P* (Eq. 17) gives an appropriate lower bound at each node of the B&B tree. It can be solved in a short time even for large instances since it is a linear programming. By relaxing the x_{ijk}'s that are still undecided in problem *P* to continuous variables with values restricted between 0 and 1 a relaxation ($LP(S_0, S_1, S_*)$) yielding a lower bound is obtained. In fact in each node a partial solution of $LP(S_0, S_1, S_*)$ is computed as the lower bound.

At each node if the solution of $LP(S_0, S_1, S_*)$ is greater than the upper bound (F^*) then the current combination of (S_0, S_1, S_*) can never lead to a solution better than F^*, since any further fixing of $x_{ijk}(ijk \in S_*)$ will only lead to objective values of at least the solution of $LP(S_0, S_1, S_*)$, and certainly not less than F^*. In this case a whole series of combinations (S_0, S_1, S_*) directly be deleted as uninteresting branch and backtracking will be done.

Upper bounding strategy

As mentioned in section 4.1 by applying algorithms I1 and I2 at the root node of the search tree in the B&B algorithm an initial upper bound is obtained. The initial upper bound is the initial value of F^*. The value of F^* may be updated in the B&B algorithm, when in optimal solution of $LP(S_0, S_1, S_*)$, for all $ijk \in S_*$, the relaxed x_{ijk}- variables turn out to be exactly 0 or 1 and the solution of $LP(S_0, S_1, S_*)$ be lower than F^*, then F^* is updated and backtracking occurs from the node. In fact in each node if a discrete feasible solution obtained and 1 and the solution of $LP(S_0, S_1, S_*)$ be lower than F^*, it means that the F^* should be updated and the related x_{ijk} and st_{ijk} are saved.

In addition to procedure described above for obtaining the upper bound, the algorithms I1 and I2 are also called at each node of the search tree to deliver an upper bound. Indeed, the algorithms I1 and I2 are implemented for $ijk \in S_*$ and if the solution of $LP(S_0, S_1, S_*)$ be lower than F^*, then the F^* value is updated and the related x_{ijk} and st_{ijk} are saved.

Branching schemes

Branching schemes in B&B are very important for the efficiency of the algorithm. In the proposed B&B algorithm for the SCC scheduling problem, branching is done by the way that the next LP relaxation problem ($LP(S_0, S_1, S_*)$) selected to be solved. For this purpose, one of the *ijk*'s in the S_* removed from S_* and put in S_1. In other words, the value of related x_{ijk} changed to 1. In this way, the new LP relaxation problem results in the next node to be solved. The selection of the *ijk* among the members of S_* is based upon the schemes that are described in this section. In fact, in the proposed B&B algorithm a new active node or depth first search strategy is used to decide upon a node from which to branch.

It should be noted that also when one of the *ijk*'s in the S_1 removed from S_1 and put in S_0, a branching was occurred too that describe later.

If S_* is empty it means that all x_{ijk}'s were already fixed, and a new feasible solution has been found for problem *P*. Hence if the conditions are satisfied the F^* value is updated and the related x_{ijk} and st_{ijk} are saved. Then backtracking will be done.

The fixing the related variable x_{ijk} first to 1 or 0 for branching is also should be determined. In the proposed algorithm the rule of first fixing the x_{ijk} to 1 and reserve the other case (0) for later while backtracking is applied.

In backtracking the fixed x_{ijk}- variable which was fixed last is considered. If it was fixed to 1 $(ijk \in S_1)$ it has still to try out fixing it to 0 (branching). Hence *ijk* is removed from S_1 and put in to S_0. If contrarily it was fixed to 0 $(ijk \in S_0)$, both cases for this variable were tried. Hence this variable leaved undecided again and backtracked further, i.e. remove *ijk* from S_0 and put it in to S_* (backtracking).

This section proposed five branching schemes which allow generating the search tree in the B&B algorithm. As indicated, the chosen *ijk* in the S_* removed from S_* and put in S_1 for branching. In fact, the nodes in the proposed B&B denoted by x_{ijk} where *ijk* is the index of the node, indicating the sequence in which the nodes are visited. Branching algorithms B1, B2, B3, B4, and B5 are designed to be used in the proposed algorithm as branching schemes.

The differences of developed branching schemes are in the order that they select the variables, x_{ijk}, being equal to value 1. This aspect is very important because it could impress the time that the final solution obtained. These branching schemes are explained respectively as follows:

- Branching scheme 1:

The branching starts from the variables related to the first machine in the EAF stage. Then the variables related to the second machine in the EAF stage will be branched, respectively. This process will continue until the last machine in the CC stage is reached. This branching scheme is introduced by algorithm B1 as follows:

Algorithm B1:

Step 1. Set j= 0, k= 0, and i= 0
Step 2. Set j= j+1.
Step 3. Set k= k+1.
Step 4. Set i= i+1.
Step 5. Whenever branching, the selection of the *ijk* among the members of S_*, is necessary in the B & B algorithm choose the related x_{ijk}- variable.
Step 6. If i< I, go to step 4.
Step 7. If k< $k_{(j)}$, set i= 0 and go to step 3.
Step 8. If j< 4, set k= 0, i= 0 and go to step 2.

- Branching scheme 2:

This branching scheme is vice versa to the previous branching scheme. In this branching rule, first the variables related to the last machine in the CC stage are branched. Then the variables related to the machine before the last machine is selected for branching. This process will continue until the first machine in the EAF stage is reached. This branching scheme is introduced by algorithm B2 as follows:

Algorithm B2:

Step 1. Set j= 4, k= $k_{(4)}$+ 1, and i= I+ 1
Step 2. Set j= j-1.
Step 3. Set k= k-1.
Step 4. Set i= i-1.

Step 5. Whenever branching, the selection of the *ijk* among the members of S_*, is necessary in the B & B algorithm choose the corresponding x_{ijk}- variable.
Step 6. If i> 1, go to step 4.
Step 7. If k> 1, set i= I+1 and go to step 3.
Step 8. If j> 1, set k= $k_{(j-1)}$+ 1, i= I+1 and go to step 2.

- Branching scheme 3:

This branching scheme is based on the charge sequence. Based on the charge sequence that are defined at the lot planning level, the variables related to the first charge in the sequence are chosen for branching, respectively from the machines of the first stage (EAF stage) to the last stage (CC stage) machines. Then the variables related to the second charge are selected for branching. This process continues until the selection of variables related to the last charge in the sequence. This branching scheme is introduced by algorithm B3 as follows:

<u>Algorithm B3:</u>
Step 1. Set j= 0, k= 0, and i= 0
Step 2. Set i= i+1.
Step 3. Set j= j+1.
Step 4. Set k= k+1.
Step 5. Whenever branching, the selection of the *ijk* among the members of S_*, is necessary in the B & B algorithm choose the corresponding x_{ijk}- variable.
Step 6. If k< $k_{(j)}$ go to step 4.
Step 7. If j< 4, set k= 0 and go to step 3.
Step 8. If i< I, set j= 0, k= 0, and go to step 2.

- Branching scheme 4:

This branching scheme is according to the charge processing time. The *ijk*'s arranged in the S_* in ascending order based on their processing time. In each iteration the last *ijk* in the S_* is removed from S_* and put in S_1 for branching. So, it means that, among the variables that are not yet branched, the variable related to the *ijk* by maximum processing time is branched first. This branching scheme is introduced by algorithm B4 as follows:

<u>Algorithm B4:</u>
Step 1. Arrange the *ijk*'s in the S_* in ascending order based on their processing time.
Step 2. Choose the x_{ijk}- variable related to the last *ijk* in the S_* whenever branching is necessary in the B&B algorithm.
Step 3. Remove the *ijk* from S_* and put it in S_1.
Step 4. If S_* isn't empty, go to step 2

- Branching scheme 5:

The last branching rule considered in the proposed B&B algorithm is based upon the scattering the charges on the machines available at each stage of the SCC system. The branching scheme by algorithm B5 is as follows:

<u>Algorithm B5:</u>
Step 1. Let C_k for k=1,…, $k_{(j)}$ be the sum of charges processing times that have already been scheduled on machine k of stage j.
Step 2. Choose the machine with the smallest C_k. Assign the next charge should be performed on the machine according to the charge sequence. The corresponding *ijk* is the next variable must be branched.
Step 3. Remove the assigned charge (*ijk*) from S_* and put it in S_1.

The performance of the five proposed branching schemes will be analyzed later in section 5. The differences of the proposed branching schemes are related to the order that the variables, x_{ijk}, being equal to value 1 in each scheme. The order of variables in the first branching scheme is based on the order of machines in stages from the EAF stage to the CC stage. In the second branching scheme this is vice versa. It means the order of variables is based on the invertor order of machines in stages from the CC stage to the EAF stage. The order of variables in the third branching scheme is according to the charge sequence that are defined at the lot planning level. In the fourth branching scheme, the order of variables for branching is based on the charge processing time. And finally, the fifth branching scheme is based on the order of the total processing time of charges on machines.

Reduce branches based upon the batch production in the CC stage

As mentioned before, one of the main attributes affecting the SCC model is batch production in the CC stage. The batch production at the CC stage is related to a job group which defined as a charges sequence that is consecutively cast on the same continues caster. Such a group of charges forms a cast, namely, a batch. The charge start time at this stage should be delayed until the charge's completion time is consistent with the start time of the next charge within the same batch. In fact, at this stage, the charges within a batch have to be processed from start to end without interruptions between charges. In this way, the processing time of a batch is equal to the sum of the charges processing times in the cast [5].

All the charges from the same batch should be processed on one machine of the CC stage consecutively while meeting given precedence constraints of charges within a batch. This characteristic of the SCC model at this stage that a charges sequence as a batch should be processed on one machine provides the opportunity to branch only the first charge in the batch not all the x_{ijk}'s variables in the batch. Then the result of the first charge of the batch in branching extended to other charges in the batch. In other words, since, all the charges from the same batch must be processed on one machine of CC stage, the result of the first charge of the batch in branching extended to other charges in the batch. By this way the number of zero-one variables in the SCC model reduces from $(i^* k(1)+ i^* k(2)+ i^* k(3)+ i^* k(4))$ to $(i^* k(1)+ i^* k(2)+ i^* k(3)+ G^* k(4))$. Since G is smaller equal than i, the number of branches is reduced. In other words, G is much smaller than i, the number of branches further reduced. Applying this method in the proposed B&B algorithm ensures reduction in branches and as a result, the computation time is reduced.

Overall flow of the proposed B&B algorithm for the SCC scheduling

As mentioned, the proposed B&B algorithm incorporates several features including initial upper bound, lower and upper bounding strategies, branching schemes, and reducing branches. The overall flow of these features in the proposed algorithm is exhibited in Fig. 3.

This algorithm can be used as an appropriate procedure to deal with the SCC model. The next section of this paper will show how the proposed algorithm operates based on the attribute specifications of the SCC model.

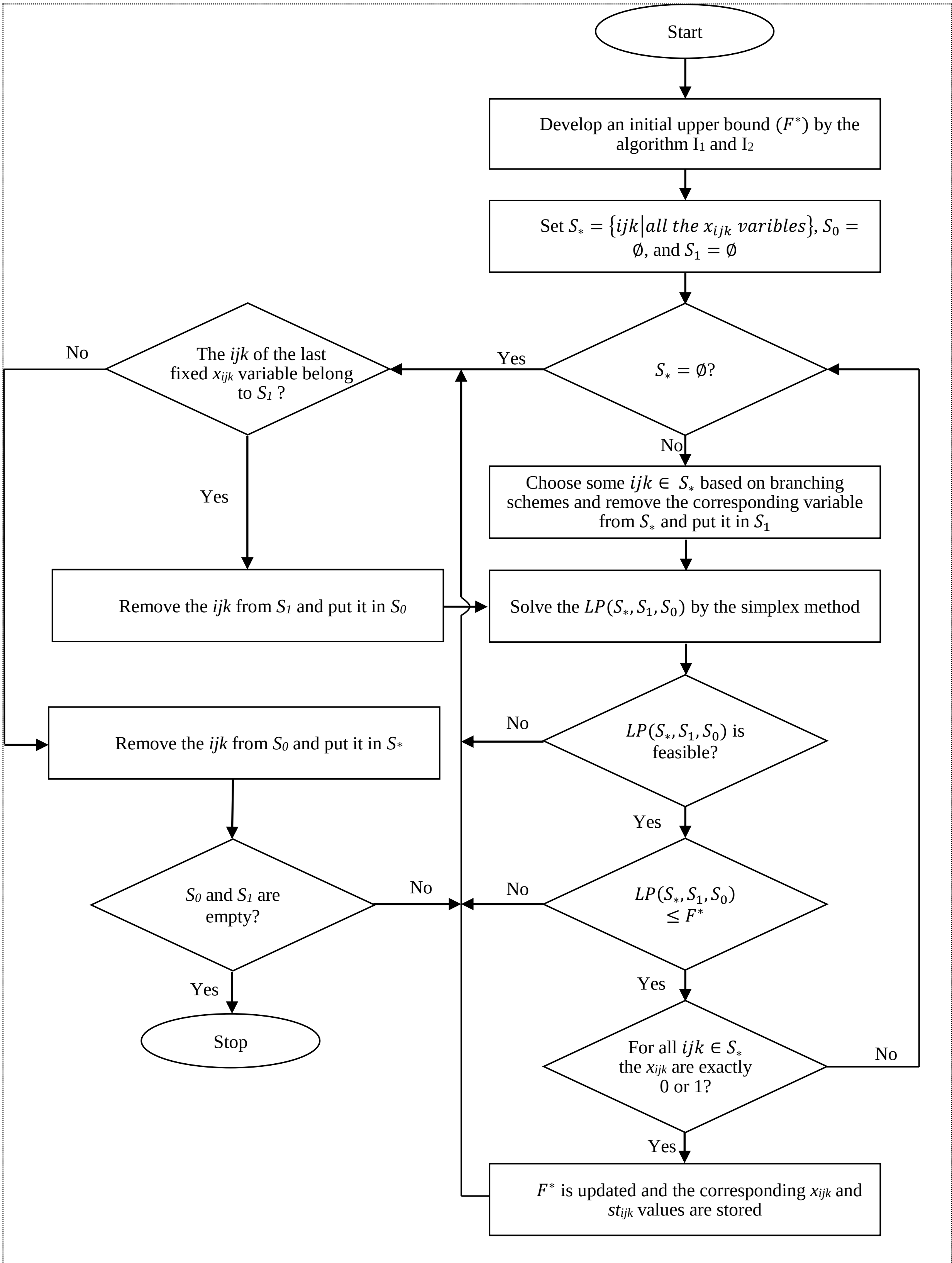

Fig. 3. Flow chart of the proposed algorithm for the SCC model.

Computational experiment

In this section, the effectiveness of the initial upper bound, lower bounding and upper bounding strategies, branching schemes, and reducing branches are investigated in problem instances of the SCC production system. The problem instances are generated based on an examination of actual production data from Iran Alloy Steel Company, the largest and most advanced iron and steel enterprise in Iran. The randomly generated problem instances show practical situations in the iron and steel industries. The computational experiment done on the instances for the different number of machines and charges. For each size of problem, 30 problems have been randomly generated. In order to reduce problem instances, every stage is assumed to have the same number of machines. However, the proposed algorithm can deal with practical problems have different number of machines for different stages. Since the maximum number of charges in each period (workday) is about 10, the computational experiments considered problems up to 10 charges. As mentioned, the steel- making plant should schedule the charges that are determined by the production planning unit. Since the plane has been designed for each workday, the planning horizon in the proposed model is set to be 1440 minutes (three eight-hour shift).

The proposed algorithm is implemented by Matlab version 10 on a personal computer with a Pentium (R) Dual- Core- 2.60 GHz CPU and 2 GB memory for problem instances in this section. Since the average percentage of all possible nodes visited in finding the optimal solution indicate an appropriate measure for the efficiency of the algorithm, the count of visited nodes is used in this section for realizing the performance of each feature of the proposed algorithm. On the other hand, this performance measure is better since it does not depend on the speed of the computer [34].

Fig. 4 shows performance of different strategies for branching in the proposed algorithm. The five proposed strategies in previous section are compared in this figure.

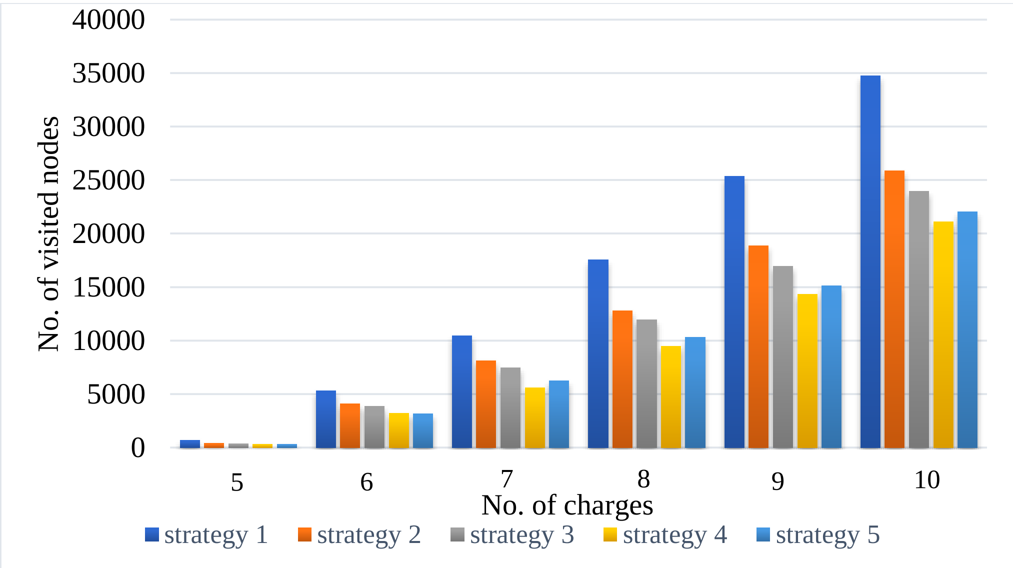

Fig. 4. Compare of proposed methods for branching

This figure shows that the performance of the proposed algorithm is affected by the strategy of branching. It is clear from the figure that the fourth scheme is the best choice for the branching method in the proposed algorithm for the SCC model. The fifth scheme is the next best strategy for branching in the algorithm. The fourth scheme is based upon the charge processing time and the fifth scheme is based upon scattering the charges on the machines available at stage 1, 2, and 3 and scattering the casts on the machines available at stage 4 of the

SCC system. In the following examples, based on the fact that the fourth strategy has better performance, this strategy is considered for branching.

Table 1 shows the number of nodes searched for obtaining the optimal solution when the initial upper bound is considered. It is clear from the table that the proposed initial upper bound reduces the number of nodes to be searched especially for the larger number of charges.

Table 1. Number of visited nodes with and without the initial upper bound (IUB)

Problem no.	No. of charges	No. of machines at each stage	Nodes visited without IUB	Nodes visited with IUB
1	5	2	385	97
2	5	3	1860	901
3	5	4	20717	26
4	6	2	4202	492
5	6	3	14492	723
6	6	4	16428	11595
7	7	2	2794	1232
8	7	3	81188	3130
9	7	4	84194	14708
10	8	2	5150	1209
11	8	3	7544	2055
12	8	4	18950	5783
13	9	2	17316	7641
14	9	3	31137	13735
15	9	4	53308	20600
16	10	2	33470	21590
17	10	3	16552	8166
18	10	4	97993	30897

Table 2 provides number of nodes visited in the problem instances when the upper bound is considered. The table shows that when the upper bound is applied in the B&B algorithm, the number of searched nodes is reduced.

Table 2. The performance of the B&B algorithm with and without the upper bound

Problem no.	No. of charges	No. of machines at each stage	Nodes visited without UB	Nodes visited with UB
1	5	2	110	97
2	5	3	942	901
3	5	4	32	26
4	6	2	534	492
5	6	3	792	723
6	6	4	12394	11595
7	7	2	1810	1232
8	7	3	3786	3130
9	7	4	18625	14708
10	8	2	1950	1209
11	8	3	2842	2055
12	8	4	6501	5783
13	9	2	11957	7641
14	9	3	17537	13735
15	9	4	25900	20600
16	10	2	28834	21590
17	10	3	9562	8166
18	10	4	47226	30897

Table 3 shows the percent reduction of zero-one variable in the proposed model by applying reduce branches method based upon the batch production in the CC stage. As mentioned before, the amount of reduction for the number of zero-one variables is ($i* k(4)- G* k(4)$). The reduction of number of zero-one variables leads to a reduction of branches. Because the relation between the number of zero-one variables and the number of branches when (h) show the number of zero-one variables is (2^h-1). Therefore, due to this exponential relation, small reduction of the number of zero-one variables leads to a significant decrease in branches.

Table 3. The performance of the B&B algorithm with applying reduce branches (RB) method

Problem no.	No. of charges	No. of machines at each stage	No. of zero-one variables	No. of zero- one variables with applying RB method	% of reduced zero- one variables with applying RB method
1	5	2	40	34	15
2	5	3	60	51	15
3	5	4	80	68	15
4	6	2	40	34	17
5	6	3	72	60	17
6	6	4	96	80	17
7	7	2	56	46	18
8	7	3	84	69	18
9	7	4	112	92	18
10	8	2	64	52	19
11	8	3	96	78	19
12	8	4	128	104	19
13	9	2	72	60	17
14	9	3	108	90	17
15	9	4	144	120	17
16	10	2	80	66	18
17	10	3	120	99	18
18	10	4	160	132	18

Finally, the performance of the proposed algorithm is considered in Table 4 for the SCC model. The number of visited nodes is negligible comparing to total number of nodes.

Table 4. The performance of the proposed B&B algorithm for SCC model

Problem no.	No. of charges	No. of machines at each stage	No. of visited nodes
1	5	2	97
2	5	3	901
3	5	4	26
4	6	2	492
5	6	3	723
6	6	4	11595
7	7	2	1232
8	7	3	3130
9	7	4	14708
10	8	2	1209
11	8	3	2055
12	8	4	5783
13	9	2	7641
14	9	3	13735
15	9	4	20600
16	10	2	21590
17	10	3	8166
18	10	4	30897

As mentioned, the problem instances up to 10 charges are considered in this paper because the maximum number of charges in each period or workday is about 10. But however, the algorithm can be used for practical instances of more than 10 charges. In order to solve problems with a very large size for example large number of charges, machines, and casts, further research is required to develop heuristic and metaheuristic methods.

Conclusion

This paper studied the SCC scheduling problem in the iron and steel industries. The SCC process is a complicated technological process that scheduling them effectively is very important for iron and steel industries. Many researchers study an SCC scheduling problem by using mathematical programming and heuristic methods, but to the best of our knowledge, B&B approach has not been reported in the literature for the SCC scheduling problem.

This paper proposed a practical method for one of the most important problems in the steel industry, means scheduling. In practical environments of the steel industry, managers have to consider on-time delivery of final products. So, the results of this paper minimize the cost function consisting of charge waiting time from operation to operation and earliness/ lateness of blooms at the end of the continuous caster.

The numerical results show that the managers of steel industries could use the proposed model and solution method of this paper to ensure on-time delivery of final products. The results of this paper are very important for the steel production industry due to special features of the SCC manufacturing process. The liquid steel is processed under a high temperature of almost above 1500 C and a large number of physical and chemical restrictions. Avoiding the large temperature drop, a waiting time between two adjacent stages, continuously processed charges on the same CC, and providing setup time to install a tundish in the CC machine before a new cast is processed on a CC are the technological constraints that considered in this paper. The results of this paper considered all mentioned features of the real-world problem through the proposed mathematical model and solution methodology. A MZONLP model was developed, considering technological constraints such as no conflict between consecutive charges processed on the same machine and also no conflict between consecutive operations for the same charge. The waiting time of charges between the processing at different stages and the total waiting time of each charge in the system are other important constraints that considered in the model. Also, the constraints result a batch production in the continuous caster stage, are considered. The objective was to ensure on time delivery of final products by minimizing a cost function including charge waiting time from operation to operation and earliness/lateness of blooms at the end of the continuous caster. The model then converted to the MZOLP model which can be solved easier and faster.

A solution methodology based on the B&B algorithm was developed to solve the problem. The proposed algorithm incorporates the initial upper bound, the lower and upper bounds, the branching schemes, and reducing branches. The computational experiments on randomly generated realistic problems in Tables 1, 2, 3, and 4 indicate that the proposed algorithm with its features could solve the proposed model quite efficiently.

The problem instances up to 10 charges are considered in this paper because the maximum number of charges in each period or workday of the case study is about 10. In order to solve problems with a very large size for example a large number of charges, machines, and casts, further research is needed to develop heuristic and metaheuristic methods. Developing the tight upper and lower bounds could be another future research to be investigated for the SCC scheduling problem. Also, providing new branching schemes and presenting the heuristic method for reducing branches are other directions for future research.

References

[1] 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- 20 (2001).

[2] Tang, L., Liu, J., Rong, A., and Yang Z. "A mathematical programming model for scheduling steelmaking- continuous casting production", European Journal of Operatoinal Research, 120, 423- 435 (2000).

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

[4] Tang, L., Luh, P. B., Liu, J., and Fang, L. "Steel-making process scheduling using Lagrangian relaxation", International Journal of Production Research, 40 (1), 55- 70 (2002).

[5] Xuan, H., and Tang L. "Scheduling a hybrid flow shop with batch production at the last stage", Computers & Operations Research, 34, 2718- 2733 (2007).

[6] Harjunkoski, I., and Grossmann, I. E. "A decomposition approach for the scheduling of a steel plant production", Computers and Chemical Engineering, 25, 1647- 1660 (2001).

[7] Gupta, J.N.D., Hariri, A.M.A., and Potts, C.N. "Scheduling a two-stage hybrid flow shop with parallel machines at the first stage", Annals of Operations Research, 69, 171- 191 (1997).

[8] Missbauera, H., Hauberb, W., and Stadler, W. "A scheduling system for the steelmaking-continuous casting process. A case study from the steel-making industry", International Journal of Production Research, 47, 4147- 4172 (2009).

[9] Sun, L. "Scheduling of Steel-making and Continuous Casting System Using the Surrogate Subgradient Algorithm for Lagrangian Relaxation", 6th annual IEEE Conference on Automation Science and Engineering, Toronto, Ontario, Canada, August 21- 24 (2010).

[10] Dao-Iei, Z., Zhong, Z., and Xiao-qiang, G. "Intelligent Optimization-Based Production Planning and Simulation Analysis for Steelmaking and Continuous Casting Process", Journal of Iron and Steel Research, International, 17(9), 19- 24, 30 (2010).

[11] Witt, A., and Voss, S. "Application of a mathematical model to an intermediate- to long-term real-world steel production planning problem based on standard software", European Journal of Industrial Engineering, 5 (1), 81- 100 (2011).

[12] Wei, L., and Liang-liang, S. "Steel-Making and Continuous/Ingot Casting Scheduling of Mixed Charging Plan Based on Batch Splitting Policy", Journal of iron and steel research, international, 19 (2), 17- 21 (2012).

[13] Gui-rong Wang, Qi-qiang Li, Lu-hao Wang "An improved cross entropy algorithm for steelmaking- continuous casting production scheduling with complicated technological routes", Journal of Central South University. 22 (8), 2998-3007 (2015).

[14] Touil Achraf, Echchtabi Abdelwahed, Bellabdaoui Adil "A simulated annealing method to optimize the order of the sequences in continuous- casting", International Journal of Mathematics and Computational Science. Vol. 1, No. 5, pp. 282- 287 (2015).

[15] Haderaa H., Harjunkoskia I., Sanda G., Grossmannc I. E., Engell S. "Optimization of steel production scheduling with complex time-sensitive electricity cost", Computers and Chemical Engineering 76, 117–136 (2015).

[16] Nastasi G., Colla V. and Seppia M. D. "A Multi-Objective Coil Route Planning System for the Steelmaking Industry Based on Evolutionary Algorithms", International Journal of Simulation Systems Science & Technologies 16, 61- 68 (2015).

[17] Armellini D., Borzonea P., Ceschiab S., Gasperob L. D., and Schaerfb A. "Modeling and solving the steelmaking and casting scheduling problem", International Transaction in Operation Research. 1–34 (2018).

[18] Li Qing, Wang Xiuying, Zhang Xiaofeng "A Scheduling Method Based on NSGA2 for Steelmaking and Continuous Casting Production Process", IFAC-Papers On Line, 51, 18, 174-179 (2018).

[19] Peng K., Pan Q., Gao L., Zhang B., Pang X. “An Improved Artificial Bee Colony Algorithm for Real-World Hybrid Flowshop Rescheduling in Steelmaking-Refining Continuous Casting Process”. Computers and Industrial Engineering. https://doi.org/10.1016/j.cie.2018.05.056, (2018).

[20] Jiang S., Zheng Z., Liu M. “A Preference-Inspired Multi-Objective Soft Scheduling Algorithm for the Practical Steelmaking-Continuous Casting Production”. Computers and Industrial Engineering, https://doi.org/10.1016/j.cie.2017.10.028, (2018).

[21] Fazel Zarandi, M. H., Dorry F. “A Hybrid Fuzzy PSO Algorithm for Solving Steelmaking- Continuous Casting Scheduling Problem”, International Journal of Fuzzy System, 20(1):219– 235 (2018).

[22] Rahal Said, Li Zukui, Papageorgiou Dimitri J. “Proactive and Reactive Scheduling of the Steelmaking and Continuous Casting Process through Adaptive Robust Optimization”, Computers and Chemical Engineering, https://doi.org/10.1016/j.compchemeng.2019.106658, (2019).

[23] Cui Haijuan, Luo Xiaochuan, Wang Yuan “Scheduling of steelmaking-continuous casting process using deflected surrogate Lagrangian relaxation approach and DC algorithm”, Computers & Industrial Engineering, 140, 106271 (2020).

[24] Long Jianyu, Sun Zhenzhong, Pardalos Panos M., Bai Yun, Zhang Shaohui, Li Chuan “A robust dynamic scheduling approach based on release time series forecasting for the steelmaking-continuous casting production”, Applied Soft Computing Journal, https://doi.org/10.1016/j.asoc.2020.106271, (2020).

[25] Portmann, M.-C., Vignier, A., Dardilhac, D., and Dezalay, D. “Branch and Bound crossed with GA to solve hybrid flow shops”, European Journal of Operational Research, 107, 389- 400 (1998).

[26] Akkan, C., and Karabat, S. “The two-machine flow shop total completion time problem: Improved lower bounds and a Branch and Bound algorithm”. European Journal of Operational Research, 159, 420- 429 (2004).

[27] Moursli, O., and Pochet, Y. “A Branch and Bound algorithm for the hybrid flow shop”, International Journal of Production Economics, 64, 113- 125 (2000).

[28] Haouari, M., Hidri, L., and Gharbi, A. “Optimal scheduling of a two-stage hybrid flow shop”, Mathematical Methods of Operations Research. 64, 107- 124 (2006).

[29] Ng, C.T., Wang, J.-B., Cheng, T.C.E., and Liu, L.L. “A Branch and Bound algorithm for solving a two-machine flow shop problem with deteriorating jobs”, Computers & Operations Research, 37, 83- 90 (2010).

[30] Ranjbar, M., Davari, M., and Leus, R. “Two Branch and Bound algorithms for the robust parallel machine scheduling problem”, Computers & Operations Research, 39, 1652-1660 (2012).

[31] Khoudi A., Berrichi A. “Minimize total tardiness and machine unavailability on single machine scheduling problem: bi-objective Branch and Bound algorithm”, Operational Research, https://doi.org/10.1007/s12351-018-0384-3, (2018).

[32] Tanaka Shunji, Tierney Kevin, Parreño-Torres Consuelo, Alvarez-Valdes Ramon, Ruiz Rubén “A Branch and Bound approach for large pre-marshalling problems”, European Journal of Operational Research, 278, 211– 225 (2019).

[33] Bunel Rudy, Turkaslan Ilker, Torr Philip H.S., Kumar M. Pawan “Branch and Bound for Piecewise Linear Neural Network Verification”. Journal of Machine Learning Research, 21, 1-39 (2020).

[34] Allahverdi, A., and Al-Anzi, F. S. “A Branch and Bound algorithm for three-machine flow shop scheduling problem to minimize total completion time with separate setup times”, European Journal of Operational Research, 169, 767- 780 (2006).