



Dynamic programming-based computation of an optimal tap working pattern in nonferrous arc furnace

Sokchomrern Ean¹ · Manas Bazarbaev¹ · Keon Myung Lee¹ · Aziz Nasridinov¹ · Kwan-Hee Yoo¹

Accepted: 8 May 2021 / Published online: 31 May 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Power is essential in nonferrous arc furnace plants for burning and melting scraps, as well as for the composition of raw materials to produce the furnace product. To ensure high-quality operation, the electric energy is controlled by changing the tap positions. However, there is no standard working pattern to determine the most effective option for changing the tap positions to obtain optimal power and product quantity. This study proposes a method to analyze and determine the working patterns in nonferrous arc furnace plants by adopting dynamic programming. To find the best objective value candidates, statistical methods were utilized to obtain the optimal values of the total elemental power and total product quantity. Moreover, if the maximum product quantity minimum electric consumption are known, the least power per product quantity (PPQ) can be easily obtained. Thus, it is reasonable to analyze the sequences of tap positions and then obtain the best PPQ using an approach of solving a recurrence problem with the widely used dynamic programming approach. We demonstrated that the proposed method suggested the working pattern of tap positions, thereby providing relatively good PPQs in comparison with the conventional method.

Keywords Nonferrous arc furnace · Tap position · Working pattern · Tap sequence · Elemental power · Product quantity · Power per product quantity · Dynamic programming

✉ Kwan-Hee Yoo
khyoo@cbnu.ac.kr

Extended author information available on the last page of the article

1 Introduction

Iron and steel are well-known materials; however, their manufacturing processes are still questionable. For the last long time, many steelmakers compromised on a longer manufacture time with high temperature to melt iron and ore for producing steel. To tackle these difficulties, **electric arc furnaces (EAFs)** were developed to support the establishment of metal [1]. According to a report by the World Steel Association in 2019 [2, 3], the EAF market is expected to witness an increase of 6.7% with respect to the revenue from 2019 (USD 1.03 billion) to 2024 (USD 1.42 billion). Notably, the steelmaking industry significantly contributes to the economy and advanced manufacturing. However, it also consumes a considerable amount of energy owing to the use of electric power for burning chemical materials and scraps. Therefore, we aim to understand how the power consumption can be efficiently controlled to increase the production as well as maintain high quality.

Several research works have attempted to find solutions for the power consumption problem to protect the EAF environment through various methods [4–7]. In the history of ferrous and non-ferrous industries, **submerged-arc furnace (SAF)**, which is a type of EAF, uses the heat generated from the material resistance as the current passes through the material, along with the heat from the electric arc between the electrode and material [6–9]. Thus, SAFs have been considered a key factor for developing melted metal products. This indicates a need to understand the background and advantages of SAF. Thus far, one of the most important features of SAF is a subcategory of EAF, i.e., the developments in the recycling of chemical materials, such as coke and slag, to produce specific products. In addition, SAF is recognized as the best method to generate metal substances [7].

Numerous researches have been conducted on topics such as tap temperature, current consumption, and power consumption in EAF including SAF. Power in EAF plants is highly essential and consumed in significant quantities in comparison to other industries [7]. Therefore, it is important to determine a method to deliver energy from the electricity source to ensure efficient furnace operations. Several relevant studies have been conducted to determine the solution to this problem. The prediction of Minoru et al. [10] introduced three methods to reduce electricity consumption in a ferroalloy firm. (1) It started from cutting off the use of the amount of slag per ton of alloy by providing different volumes of components. (2) The workers monitored the process of inserting the electrode inside the furnace. (3) Before subjecting the main and sub materials to the heating operation, each material must be checked to ensure that they are being used correctly because sensors cannot be installed in such environments owing to the high heat settings. However, the depth of the electrodes cannot be accurately known.

Kovacic et al. [11, 12] proposed a linear regression approach to define the strong relationship between materials and gases with focus on energy consumption. Moreover, genetic programming was employed based on the analysis of the most promising parameters. Consequently, this model estimated that the average energy usage could be minimized by approximately 1.16% if sufficient maintenance work is performed. Cano-Plata et al. [13] proposed a method to reduce the

loading time of scraps to the furnace container. This process was known as tap to tap of the operating time. To achieve this objective, the authors considered dynamic programming because this method can be used to boost the processes of EAFs. Fernandez et al. [14] proposed a modeling method for predicting the tapping temperature using a fuzzy interference function. Gajic et al. [15] proposed a deep learning approach to control the presence and amount of mixed materials to predict the usage of electric energy in a steelmaking industry. The data of this study were trained and tested via an artificial neural network with five-five-one hidden layers. Shyamal and Swartz [16] introduced a method to predict the changes in electricity cost to ensure the optimal utilization of electricity, gas, and melt materials. Accordingly, they proposed an online energy control approach to monitor the fluctuations in the cost of electric energy.

It is evident from the above review that most of the studies attempted to manage the input materials in a normal manner and monitor the good operation conditions with minimum energy consumption [16–23]. Normally, there are three electrodes in a nonferrous arc furnace that operate along with the transfer of energy to support the submerged-arc container while melting the scraps. For example, a good operation practice in a furnace is to maintain the tips of the three electrodes (installed inside an SAF) at the same level. However, it is difficult to obtain any information about the inside of the electric or nonferrous arc furnace environment owing to the high temperature; thus, embedded sensors or devices cannot be installed in such environments. Some researches proposed a method to adjust the position of electrodes to ensure the delivery of energy as well as the efficiency and effectiveness of the melting process. However, this method cannot be applied to all cases in a factory. The input pattern of the electric energy, quantity of materials, sequence of electrode pods, and tap positions are the factors responsible for high-quality operations [16–23]. Specifically, the workers control the tap positions while expecting minimal energy consumption because the tap positions significantly affect the power consumed in the electric furnace. This study deals with controlling the tap positions because, to the best of our knowledge, no previous study has investigated the sequence of tap positions to provide the minimum value of power per product quantity (PPQ).

However, there are no good working patterns to determine an effective option for changing the tap positions to ensure less power and high product quantity. To determine good working patterns for the tap positions, this study utilized information regarding a long sequence of tap positions obtained by performing for almost three years.

Accordingly, we focused on finding a working pattern that could enable the workers to achieve the goal by applying a dynamic programming approach [24, 25] and big data analytics [26]. Consequently, the proposed method obtained an effective working pattern of the tap sequence with minimum PPQ and short computing time.

The remainder of this paper is organized as follows. Section 2 describes the problem of this study, which practically occurs in arc furnace working environments. Section 3 explores the analysis results of the sequences of working tap positions gathered in real time from a nonferrous furnace plant. Based on the analysis of large amount of data obtained through actual field work, in Sect. 4, we propose a dynamic

programming method that provides an effective working pattern of tap positions to ensure optimal power and product quantity, and also discuss the methods for selecting a good next tap position from the current tap position considering dynamic programming. The results of our implementation are presented and analyzed in Sect. 5. The last section discusses the limitation of the proposed method and suggests future work directions.

2 Working environment and problem definition

Figure 1 illustrates an overview of the working environment and goal of the arc furnace process for producing melted metal products in the history of ferrous and non-ferrous industry [16]. The arc furnace system aims to produce large quantities of high-quality output products with less power and input materials. In other words, it should provide minimum PPQ, that is, high productivity through the operation of the system. The operation will be performed by controlling the tap positions and adjusting the material supply by the operator.

Generally, because the input materials, including the main and auxiliary raw materials, are supplied automatically through the input device, adjusting the tap greatly affects the PPQ. The core objective of this study is related to finding a sequence of tap positions to guarantee optimal PPQ. Therefore, it is important to know the definition of tap. A tap in an arc furnace is a conceptual logic that has no actual physical appearance. To some extent, the tap is essential for steelmakers to adjust the electrical current for stabilizing the performance of the burning material in the arc furnace container. Generally, the total number of tap positions is known in an arc furnace. Moreover, the tap number is different for different arc furnaces. In this study, 14 tap positions were used, which were provided by the furnace maker (labeled from 1 to 14). Naturally, the maximum value (in this case, 14) can vary depending on the furnace device.

Controlling the tap positions is the primitive actor that participates in generating the tap sequences. While a worker controls the tap positions, the process of burning raw materials and composition of electrical factors such as current, resistance, and

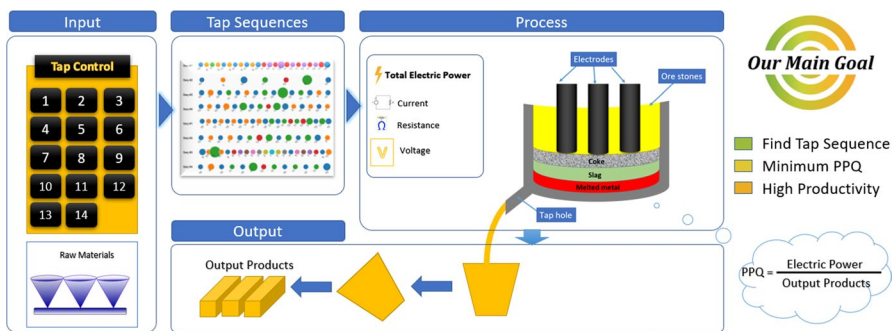


Fig. 1 Working environment and problem definition

voltage will be performed. The three electric pods in an arc furnace, as shown in Fig. 1, work as a delivery agent to transfer the power to the furnace container. Eventually, amount of products can be obtained in close dependence on the tap sequence control. These products are manufactured periodically. In this paper, a cycle is referred to as a working cycle, and the sequence of tap positions controlled by the operator during each working cycle is called a tap working pattern.

Three observations were made owing to the tap working pattern [7, 13, 16]. (1) A tap position can stay for a certain period and it significantly impacts the power consumption. (2) Tap sequence is vital for investigating the tap position control within a time series. (3) The amount of power and output products are crucial for achieving the optimal value of PPQ. These three observations contribute to high productivity. The tap sequence control aims to determine a tap sequence that can help in obtaining the minimum PPQ, which is defined as the total electric power required to produce the total amount of output products. Therefore, the operators performed their tasks considering the working conditions that can achieve this purpose. However, owing to different work situations such as amount of materials and the total consumed power, the operators adjust the tap positions differently. To consider all work situations in this study, we attempted to analyze the information related to the numerous tap sequence controls performed by operators thus far. Based on the analysis results, we proposed a dynamic programming method to obtain the optimal tap sequence, i.e., an optimal tap working pattern.

3 Analysis of tap working patterns

The movement of tap positions in an electric furnace has some constraints. Figure 2 demonstrates the possible movements of a tap position after 1 s. For instance, assume that the current tap is in position i . Accordingly, there are three possible directions that can be attained after 1 s. First, the current tap can move to the previous tap position, i.e., position $(i-1)$. Second, the current tap can move to the next tap position, i.e., position $(i+1)$. Finally, the current tap is likely to stay still at the same position after 1 s. Thus, the tap position movement pattern is different for each working cycle.

The tap position sequence is presented to provide a general overview of the tap movement trend with respect to the time series data from the firm. Before further examining this sequence, some key terms must be introduced. The sequence, here,

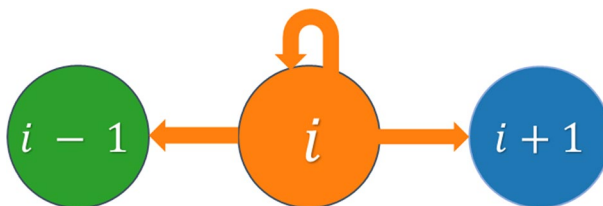


Fig. 2 Constraints of tap movement

easier to understand the tap sequence in each working cycle, the visualization of the tap sequences is drawn as shown in the Fig. 4.

Figure 4 presents the tap sequences of a specific product (Product A) manufactured in one day. The first column presents information regarding the sequence number. There are four sequences in this example (from 1 to 4) for the product name specified in the second column. The third and fourth columns denote the start time and end time of the working cycle, respectively. The WCT is defined as the end time minus the start time. The fifth column represents the total electric power (TEP) consumed during one working cycle with respect to the changes and control of tap by the operator in the working area. The sixth column is the amount of total product quantity (TPQ), which is the output of the WCT. The next column is the PPQ, which is the result of the total power divided by TPQ. The final column consists of the tap sequence, which serves as the tap movement pattern in the conceptual framework.

Figure 5 reveals the big picture that demonstrates the characteristics of the tap sequence such that the visualization of the objects can be utilized to represent the main objective of the tap working patterns. For instance, the circles denote the tap number. The size of each circle is not the same because the stay time and duration of each tap position are different. Therefore, the bigger the circle, the longer the stay time of the tap position. Interestingly, the different colors also help in effectively distinguishing the tap position number. Each tap in the tap sequence number has three values (the tap position number, stay time, and electric power). Here, the stay time denotes the time spent at a tap position and electric power denotes the amount of power consumed during the stay time at the tap position. The values at each tap position can be obtained from the data gathered in real time in seconds from the programmable logic controller (PLC) in an arc furnace. The real-time large amount of data from the time series are employed to obtain the tap information every second from the starting time to the ending time within one working cycle. Moreover, the inclusion of electrical consumption and product amount in the historical data works as the source for constructing the model of this study.

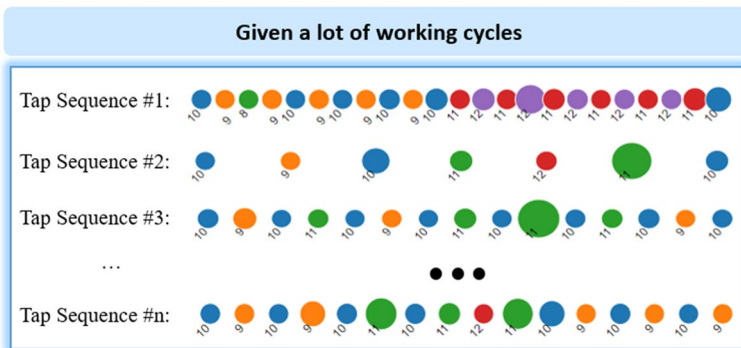


Fig. 5 Tap sequences of numerous working cycles

For analysis purposes, the tap sequence plays a crucial role in demonstrating the initial step for analyzing the tap working patterns from several methods, such as statistical analysis and advanced computer algorithms. The analysis was verified by considering a working cycle that should have at least one tap position. Thus far, the criteria for selecting the tap positions were determined by the operator manually. Moreover, the data were gathered from multiple sources at various time points during and after the operation. For instance, the tap position numbers were received from the PLC in an arc furnace, while the number of products and raw materials were recorded by the steel worker. It should be noted that a critical part of this analysis is determining how to match the data recorded by users on paper with the digital real-time data obtained from the PLC.

To analyze the tap sequence pattern, first, the nature of the tap position should be investigated with respect to the frequency of the tap position, stay time of the tap position, connection between each tap, and the amount of consumed electric energy with respect to the output of product quantity and specific time frame of operation. The tap sequence has a unique characteristic related to the movements. There can be three movements after 1 s, i.e., the next tap position ($i + 1$), previous tap position ($i - 1$), and the same position i . The notation i denotes the number of tap position.

Figure 6 illustrates several tap positions in a tap position sequence within one working cycle. In the top part of the figure, the straightforward visualization draws attention to the series of tap position nodes presented by circles with various colors for different tap position numbers. This figure also provides information regarding the source of tap sequence analysis. The first row presents a list of time (in seconds), denoted by j . For instance, these times, represented in seconds, start from one, until the final time for one working cycle. The second row denotes the tap position number that is used to represent the tap number from 1 to 14; its notation is $TP_{i,j}$, where TP stands for tap position, and i and j are the tap position number and the time elapsed from the starting time, respectively. The third row denotes the elemental power, $EP_{i,j}$, consumed by tap position i at j th second from the starting time. The data related to the tap position, $TP_{i,j}$, and elemental power, $EP_{i,j}$, were retrieved in real time from an arc furnace machine. The last row denotes the product quantity, $PQ_{i,j}$, obtained by tap position i at j th second from the starting time. Because the product quantity cannot be obtained in real time, in this study, we calculated the product quantity using the total amount produced during the working cycle. Because

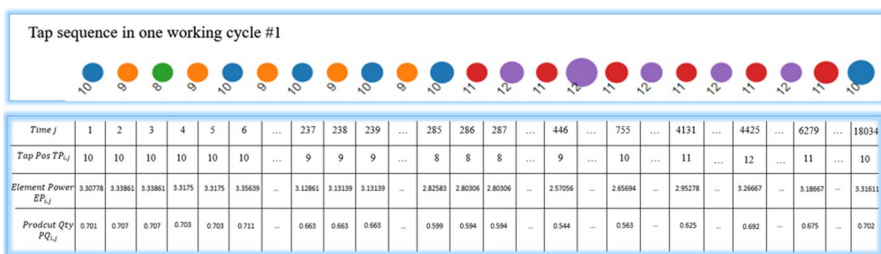


Fig. 6 Tap sequence and constraints of movement

the consumed electric power $EP_{i,j}$ for tap position i at j th second from the starting time is obtained from an arc furnace in real time, the obtained product quantity $PQ_{i,j}$ at this time can be calculated as follows:

$$PQ_{i,j} = TPQ * EP_{i,j} / TEP \tag{1}$$

Figure 6 also highlights the characteristic of tap movement with the normal situation. If there is another movement that is different from the previous one, the new movement is considered to be abnormal. For example, assume that tap position nine moves to tap number twelve after 1 s. In such a case, a system error may occur in some part while generating the data. Together from results of the analysis relating to the working tap position sequence, we will provide a preliminary idea to support the selection of the best working patterns in the next step. It should be noted that the number of working cycles per day can be between 1 and 5. In this study, a total of 1055 working cycles were recorded within a period of 11 months for data collection. The dataset was extracted from October 2018 to August 2019.

4 Finding an effective tap working pattern using dynamic programming

4.1 Proposed dynamic programming

After analyzing the working patterns from large amount of previous real-time data, the analysis results were stored in a relational database for future use. As shown in Fig. 7, for a tap sequence in a working cycle, the movement of a tap position can be formulated in the form of a table, which is vital for accessing specific elements to track the pattern. The starting tap position number is 10 and the duration of the working cycle is 23,339 s. The red polyline represents the tap sequence. The table has multiple elements that contain useful information. The top horizontal row shows the sequence of time series in seconds, while the leftmost vertical column presents the tap position number. Each tap sequence has the starting time and ending time.

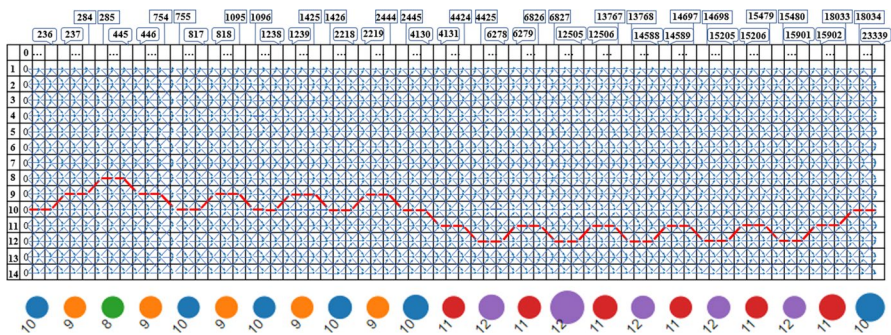


Fig. 7 Original trend of a working pattern with a working cycle time, WCT=23,339 s

To provide a universal method at a level that does not lack generality, the starting time is rewritten to the number of times in seconds in ascending order, from zero to the final time of one working cycle.

This study considered all tap sequences for more than 1,055 working cycles within a period of 11 months for data collection. The data obtained in real time constituted large amount of data that underwent processing techniques of the data, such as outlier and missing value processing [24]. Eventually, the data information including the electric power and product quantity was stored in an element in the table, i.e., tap position i and second j . Here, i denotes the value of tap positions from 1 to 14, and j is one of total seconds from the starting time 0 .

Now, let us consider the objective of the problem. We aim to determine a working pattern for the tap sequence that minimizes the following function:

$$\min_{wp_i, i=1, \dots, TWP} PPQ_{wp_i} = \frac{TEP_{wp_i}}{TPQ_{wp_i}} \tag{2}$$

where wp_i denotes a working pattern of a tap sequence to give the optimal PPQ; TWP is the total number of possible working patterns; TEP_{wp_i} and TPQ_{wp_i} represent the TEP and TPQ at wp_i , respectively; and PPQ_{wp_i} is defined to be $\frac{TEP_{wp_i}}{TPQ_{wp_i}}$ at wp_i .

Assume a WCT of 23,339 s and $TapSize$ is the number of tap positions in an electric furnace. There are a total of $TapSize * 3^{(WCT=23,339)}$ possible tap sequences because that the constraint of tap position movement is needed to consider going along the way. In other words, tap position i can either become $(i + 1)$, $(i - 1)$, or i in the next step. Figure 7 shows the case when $WCT = 23,339$; the red polyline represents one case among the $3^{23,339}$ working tap position sequences. Because it takes exponential time to perform the best working tap position sequence, it is interesting to check the possibility of selecting different paths from 1 s to the next. By doing that, the optimal methodology is needed to solve this problem.

In this study, we proposed a dynamic programming approach to find the optimal tap working pattern that can provide high efficiency in terms of energy usage and the maximum product quantities. Dynamic programming is a design method that is generally used to solve a given problem owing to a series of decisions [25, 26]. This problem, i.e., the inability of making a series of stepwise decisions that result in an optimal decision order, can be solved by implementing every possible decision order. In other words, after enumerating all decisions in each step, we can select the best among them. In dynamic programming, the optimal decision order has the characteristic that the decision at the current time should constitute the optimal decision order with respect to the state due to the previous decision, irrespective of the initial state and decision. Therefore, it can be observed that the optimal solution to a given problem via dynamic programming is always obtained by considering all cases sufficiently. The most outstanding theme of the approach is dividing the problem into many sub problems. Importantly, the outputs of the subproblems have been stored and reused next time. Thus, various optimization problems, such as the shortest path and 0–1 knapsack, can be solved via dynamic programming approach [25].

Table 1 Notations

Term	Description
$PPQ[i,j]$	PPQ calculated at tap position i from the starting time to time j
$TEP[i,j]$	TEP at tap position i from the starting time to time j
$TPQ[i,j]$	TPQ at tap position i from the starting time to time j
WCT	A working cycle time
$BC_EP[i,j]$	The best candidate of elemental power at tap position i and time j
$BC_PQ[i,j]$	The best candidate of product quantity at tap position i and time j
$STP[i,j]$	Tap position at time $j-1$ to be moved to tap position i and time j
TAD	A database storing the analysis results of tap working patterns

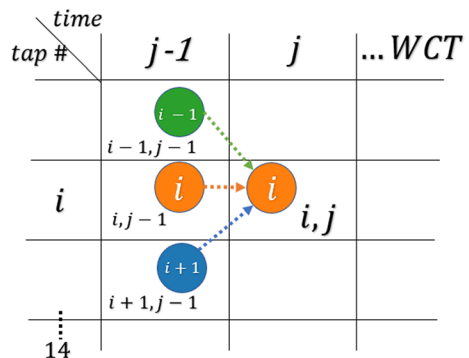
With respect to the problem of this study, we attempted to use this approach to select a universal pattern for the working tap sequence to investigate tap position movements in an EAF environment. Before implementing the dynamic programming approach, we analyzed several tap working patterns in an electric furnace, and the amount of power consumed at each tap position was stored in the analysis database (TAD).

To apply the dynamic programming approach, the following notations were defined as listed in Table 1.

To obtain the minimum $PPQ[i,j]$ at tap position i from the starting time to time j , first, $TEP[i,j]$ and $TPQ[i,j]$ must be calculated by considering three previous results and three tap positions ($(i-1)$, i , and $(i+1)$) from time $j-1$ after 1 s, as shown in Fig. 8. The value of k in Eqs. (3) and (4) indicates either of the tap positions i , $i-1$, or $i+1$.

Before obtaining the best $PPQ[i,j] = TEP[i,j] / TPQ[i,j]$, two values, i.e., $TEP[i,j]$ and $TPQ[i,j]$ must be calculated by the following recursive equations (Eqs. (3) and (4) from the TEP and TPQ obtained from previous stages, respectively). Basically, when an initial tap position is located at initial_tap, $TEQ[\text{initial_tap}][0] = 0$, $TPQ[\text{initial_tap}][0] = 0$, $PPQ[\text{initial_tap}][0] = 0$, and $STP[\text{initial_tap}][0] = \text{initial_tap}$ will be assigned. In other cases, $TEQ[i][j] = \text{Infinity_Value}$, $TPQ[i][j] = \text{Small_value}$, $PPQ[i][j] = TEP[i][j] / TPQ[i][j]$, and $STP[i][j] = \text{NaN}$ will be assigned.

Fig. 8 Decision making to select the best $PPQ[i,j]$ at tap position i from the starting time to time j



$$TEP[i,j] = \begin{cases} \text{Infinity Value} & \text{otherwise} \\ TEP[k,j-1] + BC_EP[k,j-1] & \text{if } j > 0 \text{ and } k = i-1, i, i+1 \end{cases} \tag{3}$$

$$TPQ[i,j] = \begin{cases} \text{epsilon} & \text{otherwise} \\ TPQ[i+k,j-1] + BC_PQ[i+k,j-1] & \text{if } j > 0 \text{ and } k = i-1, i, i+1 \end{cases} \tag{4}$$

The pseudocode to solve our given problem by applying dynamic programming approach is provided, as shown in the Finding_Optimal_Tap_Working_Pattern algorithm. The algorithm is based on the bottom-up approach to investigate the appropriate candidates that align with the objective of determining the minimum TEP and maximum product quantity to provide better PPQ results. The input parameters include the initial tap position (initial_tap), WCT, and total tap position number (TapSize), provided by an arc furnace. Additionally, a database (TAD) that stored the analysis results of several working patterns was provided as the input. These input parameters are crucial for implementing the algorithm. The output of the algorithm includes the TEP, TPQ, and PPQ from the start time to every second in the given WCT; finally, the best PPQ and best tap sequence (STP) during the WCT can be obtained.

Figure 8 demonstrates the steps for producing the outputs from the input parameters. The related data structures are assigned an initial value through steps 1–8 for every tap position (TapSize) during a WCT. Next, we focus on obtaining the outputs at tap position i and time j (steps 9–10). Steps 11–25 of the algorithm determine the best from the three possible candidates at tap position i and time j ; these three candidates are processed in step 11 (for $k = -1$ to 1). First, the best product quantity and best elemental power were selected from the numerous candidates at tap position $(i-1)$ with time $(j-1)$ at step 12 and step 13, respectively. Subsequently, steps 14–20 were used to check the minimum, localMinPPQ, from the three PPQs to get the minimum index, localMinIndex. The localMinIndex will be used to represent the final best product quantity and best elemental power to add with the TPQ at tap position $(i-1)$ with time $(j-1)$, as well as TEP at tap position $(i-1)$ with time $(j-1)$, respectively. The work will be done by considering the last two cases. The first case ($k=0$ in step 11) is doing the same for tap position i and time $j-1$, the second case ($k=1$ in step 11) for tap position $(i+1)$ and time $(j-1)$. After determining the best candidates (steps 14–20), steps 22 and 23 add the results with the previous TPQ and TEP, respectively. Then, the optimal PPQ at step 24 denotes the result of the TEP divided by TPQ. Further, the selection tap position is stored (step 25).

This operation is repeated for all remaining tap positions and times. The details of the algorithm are described in Finding_Optimal_Tap_Working_Pattern.

Finding_Optimal_Tap_Working_Pattern. Finding an effective working pattern *STP* of tap sequence to provide the best *PPQ* from an initial tap position *initial_tap* during a given working time *WCT*.

Input:

- (1) *initial_tap*: initial tap position at time 0
- (2) *WCT*: given working time
- (3) *TAD*: database storing the analysis results of several tap working patterns.
- (4) *TapSize*: total tap position number provided by an arc furnace.

Output:

- (1) *STP*: sequence of optimal tap positions calculated by the algorithm
- (2) *PPQ*: power per product quantity obtained with the *STP*

BEGIN

```

// TapSize: fixed number of tap positions.
// localMinPPQ: local minimum PPQ.
// localMinIndex: index to localMinPPQ.
// initialize all data structures
1. FOR j = 0 to WCT // iterate the time from 1 second to the given T
2.   FOR i = 1 to TapSize // iterate from the first to the last tap
3.     TEQ[i][j] = Infinity_value;
4.     TPQ[i][j] = Small_value;
5.     PPQ[i][j] = TEP[i][j]/TPQ[i][j];
6.     STP[i][j] = NaN;
7. TEQ[initial_tap][0]=0; TPQ[initial_tap][0]=0;
8. PPQ[initial_tap][0]=0; STP[initial_tap][0]= initial_tap;
9. FOR j = 1 to WCT // iterate the time from 1 to the given T
10.  FOR i = 1 to TapSize // iterate from the first to the last tap
11.    FOR k = -1 to 1 // ad-hoc index for storing the candidate
12.      // obtain TPQ candidate for each step
13.      tpq = TPQ[i+k][j-1] + BC_PQ[i+k][j-1];
14.      // obtain candidate of TEP for each step
15.      tep = TEP[i+k][j-1] + BC_EP[i+k][j-1];
16.      // obtain PPQ candidate for each step
17.      IF (k== -1) {
18.        localMinPPQ = tep/tpq;
19.        localMinIndex= -1;
20.      } ELSE IF (localMinPPQ > tep/tpq){
21.        localMinPPQ = tep/tpq;
22.        localMinIndex= k;
23.      }
24.    END k
25.    // calculate the TPQ
26.    TPQ[i][j]=TPQ[i+localMinIndex][j-1] + BC_PQ[i+localMinIndex][j-1];
27.    TEP[i][j]=TEP[i+localMinIndex][j-1] + BC_EP[i+localMinIndex][j-1];
28.    PPQ[i][j]=TEP[i][j]/TPQ[i][j];
29.    STP[i][j]= STP[i+localMinIndex][j-1] + localMinIndex;
30.  END i
31. END j

```

4.2 Selecting best candidate in proposed dynamic programming

The Finding_Optimal_Tap_Working_Pattern algorithm in Sect. 4.1 highlights an important point related to the selection of the best PPQ by finding the best product quantity (BC_PQ) and best elemental power (BC_EP) from the several candidates stored in the tap working pattern analysis database, i.e., TAD.

Now, consider $BC_PQ[i][j]$ and $BC_EP[i][j]$ at tap position i and time j . To select $BC_PQ[i][j]$ and $BC_EP[i][j]$, we must explore the elemental powers and product quantities stored at the tap position and time. The values can be obtained by analyzing the experimental candidates, which were obtained through real works until now. Assume that a tap working pattern $WP=(TP_0, TP_1, \dots, TP_{wct})$ during a WCT is given, where TP_j is a tap position at j th second. As mentioned in Sect. 4.1, note that the TPQ produced during the WCT is provided manually; however, the tap positions and elemental powers are gathered in real time every second. Let $ep[i][j]$ and $pq[i][j]$ be the elemental power and product quantity at tap position i and time j , respectively. Then, $pq[i][j]$ is calculated according to Eq. (5). The analysis results of the tap working patterns are stored in an analysis table, TAD .

$$pq[i][j] = \frac{ep[i][j] * TPQ}{TEP} \tag{5}$$

As illustrated, many tap working patterns are used to ensure better results in selecting BC_EP and BC_PQ. When the tap working patterns are considered, several tap analysis results at tap position i and time j may be stored in the TAD . Figure 9 illustrates the matrix containing the elemental powers and product quantities at tap position i and time j , highlighted in the interception of x-axis i and y-axis j in the matrix. Every candidate for BC_EP and BC_PQ may be stored in the analysis table of large amount of data. Table 2 presents information regarding the analysis at tap position 11 and 100 s on 19 elemental powers, 19 product quantities, and 19 PPQs. It is important to select the best ones from $TAD[i][j]$.

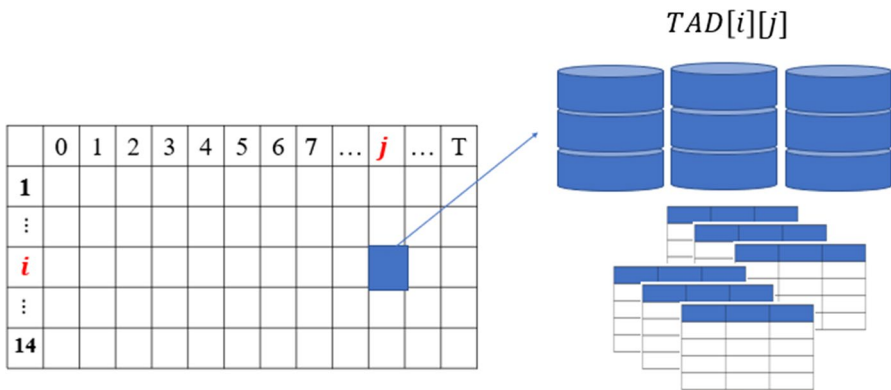


Fig. 9 Decision making to select BC_EP and BC_PQ

Table 2 Sample data stored in TAD for tap position 11 and 100 s

No	Tap position	Time (s)	Elemental power (ep[$i=11,j=100$])	Product quantity (pq[$i=11,j=100$])	Power per product quantity (ppq[$i=11,j=100$])
1	11	100	12,065	11,544.3	1.05
2	11	100	11,899	13,708.9	0.87
3	11	100	12,460	19,481	0.64
4	11	100	12,023	14,864.9	0.81
5	11	100	12,491	11,891.9	1.05
6	11	100	11,065	14,864.9	0.74
7	11	100	11,975	13,750	0.87
8	11	100	12,258	13,095.2	0.94
9	11	100	12,029	14,404.8	0.84
10	11	100	12,036	11,700	1.03
11	11	100	12,188	13,000	0.94
12	11	100	12,951	14,300	0.91
13	11	100	12,546	15,151.9	0.83
14	11	100	12,809	15,151.9	0.85
15	11	100	11,887	12,623.4	0.94
16	11	100	12,331	11,922.1	1.03
17	11	100	12,826	15,428.6	0.83
18	11	100	12,471	14,026	0.89
19	11	100	12,517	14,054.8	0.89

Next, we examine Eq. (2). Our objective is to get the minimum PPQ in each step by applying the dynamic programming approach. Accordingly, we first attempted to use the most basic statistical methods [27], i.e., min, max, average, and deviation. All relevant factors were considered as the input parameters to find the best path of tap position movement. This research aims to realize the least power usage along with the maximum output values. Thus, the calculation of the minimum of elemental power, minimum deviation of tap position, and maximum product quantity were considered.

BC_MinMax(i,j), which is the first statistical method, determines the best candidates of elemental power and product quantity using min and max approaches. First, the input parameters include the tap position number, number of seconds, tap analysis data, and number of elements. The algorithm executes a query to filter the observation that contains the minimum elemental power and maximum product quantity, and is eventually subjected to the least value of PPQ. Thus, the minimum elemental power and maximum product quantity in $TAD[i,j]$ are extracted successfully.

BC_MinMax(i,j). Finding the minimum BC_EP and maximum BC_PQ at tap position i and j second

Input:*tap position i and time j**TAD[i,j]*: elemental powers and product quantities stored at tap position i and time j *daily_ppq*: a PPQ recorded by a steelworker in a daily report**Output:***BC_EP[i,j]*: the best candidate of elemental power at tap position i and time j *BC_PQ[i,j]*: the best candidate of product quantity at tap position i and time j **BEGIN**

// select the minimum BC_EP

BC_EP[i,j] = minimum among elemental powers in *TAD[i,j]*

// select the maximum BC_PQ

BC_PQ[i,j] = maximum among product quantities in *TAD[i,j]***END**

The second statistical method is BC_Average(i,j), which finds the best candidates of elemental power and product quantity by selecting the average approach. Primarily, the input parameters include the tap position number, number of seconds, tap analysis data, and number of elements. Second, the algorithm runs an average function to get the average elemental power and average product quantity. Next, the selected record must fulfill two criteria; first, the elemental power value should be less than or equal to the average elemental power; the second criterion focuses on the product quantity value that is greater than or equal to the average product quantity. Most importantly, the record must guarantee the minimum PPQ.

BC_Average(i,j). Finding the BC_EP and BC_PQ that are close to the average values at tap position i time j

Input:*tap position i and time j**TAD[i,j]*: elemental powers and product quantities stored at tap position i and time j *daily_ppq*: a PPQ recorded by a steelworker in a daily report**Output:***BC_EP[i,j]*: the best candidate of elemental power at tap position i and time j *BC_PQ[i,j]*: the best candidate of product quantity at tap position i and time j **BEGIN**

// calculate the average elemental power

AVG_EP[i,j] = average of elemental powers in *TAD[i,j]*

// calculate the average product quantity

AVG_PQ[i,j] = average of product quantities in *TAD[i,j]*// select an elemental power whose value approximates to *AVG_EP[i,j]**BC_EP[i,j]* = elemental power that is nearest to *AVG_EP[i,j]*// select a product quantity whose value approximates to *AVG_PQ[i,j]**BC_PQ[i,j]* = product quantity that is nearest to *AVG_PQ[i,j]***END**

The third statistical method is BC_Deviation(i,j), which employs the deviation approach to find the best candidates for the elemental power and product quantity. First, the tap position number, number of seconds, tap analysis data, and number of elements serve as the input parameters. Further, ppq, which is recorded by the steelworkers, is used as a key indicator to find the smallest deviation. Therefore, the algorithm calculates ppq in TAD[i,j]. Then, daily_ppq subtracts ppq to get the deviation value. The record with the smallest deviation and positive value is selected.

BC_Deviation(i,j). Finding the BC_EP and BC_PQ that have the smallest deviation between ppq and *daily_ppq* at tap position i and time j

Input:

tap position i and time j

TAD[i,j]: elemental powers and product quantities stored at tap position i and time j

daily_ppq: PPQ recorded by a steelworker in a daily report

Output:

BC_EP[i,j]: the best candidate of elemental power at tap position i and time j

BC_PQ[i,j]: the best candidate of product quantity at tap position i and time j

BEGIN

// select elemental power with the smallest deviation ppq

BC_EP[i,j] = elemental power that contributes to obtain the smallest deviation |ppq- *daily_ppq* |

// select product quantity with the smallest deviation ppq

BC_PQ[i,j] = product quantity that contributes to obtain the smallest deviation |ppq- *daily_ppq* |

END

From now, let us analyze the computation complexity of our proposed Finding_Optimal_Tap_Working_Pattern algorithm. Since Step (1) through Step (6) are repeated two for loops, the computational complexity of the steps is $O(WCT*TapSize)$. From Step (9) to Step (27), three for loops are repeated, but because k ($=-1,0,1$) is repeated three times, so that it is repeatedly executed as much as $3*WCT*TapSize$. In each loop, BC_PQ[i,j] and BC_EQ[i,j] must be selected. The selection time complexity is dependent on the number of items stored in TAD[i,j]. Assuming that the number is n , the BC_PQ[i,j] and BC_EQ[i,j] can be obtained in $O(n)$ time even if any of the three proposed methods is used. So the total computation complexity of the algorithm is $O(WCT*TapSize*N)$, where N is the maximum number of items in TAD[,]’s.

5 Experimental results and performance evaluation

5.1 System environment

This paper describes the methods and techniques used in this study to select an appropriate working pattern that is subjected to the optimal value of PPQ. We used a computer with the following specifications for the experiments: CPU Intel® Core™ i7-6700, 3.40 GHz, RAM 32 GB, NVIDIA GeForce 9800 GT graphics card,

Windows 10 operating system, and Jupyter Notebook, an integrated development environment for Python and IntelliJ for Java [28–30]. Moreover, MariaDB was used to store the enormous amount of data.

5.2 Data preparation

The data contained in the dataset were generated by a steelmaking firm. A total of 24.59 million records, from October 2018 to August 2019, were utilized. Interestingly, the data were obtained from two types of sources. The first type of data was extracted from daily reports available in the form of hard copies. It consisted information related to the product name (PN), production date, quantity of output, mixed material name, quantity of raw materials, tapping periods, and amount of power consumption in one day. However, the data recorded by humans may have inconsistency issues; for instance, the number of products, amount of power, and production time. This is responsible for the numerous steps involved in this process. Therefore, the data must be preprocessed and verified with the data operators. The second type of data was obtained from a PLC. The PLC transfers the data from the machines in a factory to the database at the end of the day, approximately midnight. All records are arranged according to their timestamps every second.

Tables 3 and 4 present 21 datasets from the 1,397 working cycles with respect to specific products. The data were collected from October 20, 2018 to August 31, 2019. As evident in the tables, there are 3–5 working cycles in a day, and each WCT is calculated by subtracting the working start time (WST) from the working end time (WET). Generally, the minimum, maximum, and average of the working cycle duration are 6839, 65,990, and 23,238 s, respectively. During each working duration with the sequence number (SN), the TEP is consumed to produce the TPQ for a specific PN such that the PPQ is obtained by dividing TEP by TPQ. These operations are performed according to the STP during the WCT, which is controlled by an employee. The STP is called a working pattern. In our experimental data, we used a tap controller that can locate a tap at one of the $TapSize = 14$ positions. According to the tap control by the operator, the electric furnace provides the tap number and the element power consumed at the tap number at each second in a WCT. With this information, we provided three values [a, b, c] for each tap in a working pattern, where a denotes the tap number, b is the time remaining on the tap, and c is the PPQ on the tap using Eq. (6). Table 4 presents the detailed tap sequence information of the 21 working patterns summarized in Table 3.

5.3 Proposed working pattern results

With respect to the analysis results of large amount of data illustrated in the previous section, the Finding_Optimal_Tap_Working_Pattern algorithm was applied to find an effective working pattern while reflecting its profound effect on the PPQ. Before applying the proposed algorithm, the working patterns with the data, as demonstrated in Tables 3 and 4, were preprocessed according to each PN. The details of the preprocessing are described in Sect. 3. Based on the preprocessing results, the

Table 3 Information regarding the working patterns

No	PN	SN	WST	WET	TEP	TPQ	PPQ
1	Product D	1	2019-06-01 9:59:00	2019-06-01 16:02:59	61,348.50	11,049.94	5.55
2	Product D	2	2019-06-01 16:03:00	2019-06-01 22:05:59	62,235.65	9099.92	6.84
3	Product D	3	2019-06-01 22:06:00	2019-06-01 23:59:59	18,321.75	9100.01	2.01
4	Product D	1	2019-06-03 9:20:00	2019-06-04 0:10:59	62,292.87	10,400.07	5.99
5	Product D	2	2019-06-04 0:11:00	2019-06-04 6:07:59	58,594.97	9600.09	6.10
6	Product D	1	2019-06-04 6:08:00	2019-06-05 0:16:59	93,001.25	9843.82	9.45
7	Product D	2	2019-06-05 0:17:00	2019-06-05 6:04:59	54,464.51	11,156.22	4.88
8	Product A	1	2019-07-12 6:17:00	2019-07-12 23:56:59	101,294.87	30,779.13	3.29
9	Product A	2	2019-07-12 23:57:00	2019-07-13 6:19:59	71,284.38	37,221.24	1.92
10	Product A	1	2019-07-13 6:20:00	2019-07-13 12:06:59	61,752.64	28,556.19	2.16
11	Product A	2	2019-07-13 12:07:00	2019-07-13 18:01:59	60,940.31	22,715.24	2.68
12	Product A	3	2019-07-13 18:02:00	2019-07-13 23:59:59	60,314.74	23,364.22	2.58
13	Product A	1	2019-07-15 6:15:00	2019-07-15 12:03:59	66,242.02	28,838.27	2.30
14	Product A	2	2019-07-15 12:04:00	2019-07-15 18:00:59	64,615.54	26,155.63	2.47
15	Product A	3	2019-07-15 18:01:00	2019-07-16 0:53:59	74,942.93	28,167.70	2.66
16	Product A	4	2019-07-16 0:54:00	2019-07-16 6:19:59	65,663.62	28,838.39	2.28
17	Product A	1	2019-07-16 6:20:00	2019-07-16 23:28:59	100,721.56	33,877.54	2.97
18	Product A	2	2019-07-16 23:29:00	2019-07-17 6:11:59	77,778.51	49,122.44	1.58
20	Product A	1	2019-07-17 6:12:00	2019-07-18 0:03:59	114,003.58	30,778.66	3.70
21	Product A	2	2019-07-18 0:04:00	2019-07-18 6:32:59	72,712.63	43,221.12	1.68

proposed dynamic programming algorithm was implemented with input parameters such as the PN, initial tap position (*initial_tap*), WCT, and one of the three proposed methods for selecting good BCs in Sect. 4.2. Consequently, the proposed algorithm yielded TPQ, TEP, and PPQ as the outputs in Table 5, and an effective tap sequence in Table 6. Additionally, the execution time (ET) of this algorithm is presented.

For example, consider the data in the first and seventh rows of Tables 5 and 6. In the first row, the input data include Product A, 11 initial tap position, 4,000 s as a WCT, and BC_Deviation(). The output results are 9365.10 TPQ and 3745.12 TEP; thus, the PPQ is 2.50. Additionally, the ET is 148 s. The effective tap sequence obtained from the algorithm with the row data in Table 5 is presented in the first row of Table 6. Let us interpret the tap sequence output in detail. Let us begin at tap position 11, stay at the same tap position for 5 s, and then move to tap position 12. After staying at tap position 12 for 1 s, move to tap position 11 again. It can be observed that after staying at tap position 11 for 61 s, it moves to tap position 12 again. The movement of this tap will repeat for 4,000 s, as shown in the first row in Table 6. Here, the data [5, 11] in the tap sequence implies that the tap stays at tap position 11 for 5 s.

Similarly, when interpreting the contents of the seventh row of Table 5, Product C, initial tap position 11, WCT of 8,000 s, and BC_MinMax() are given as inputs. The output results are 3500.59 TPQ and 1263.89 TEP; thus the PPQ is 2.77. Additionally, the ET is 546 s. Initially, the tap stays at tap position 11 for

Table 4 Tap position sequences per working cycle

No	STP
1	[10, 481, 3.057], [9, 2153, 2.46], [10, 202, 2.885], [11, 11,883, 3.155], [10, 2953, 3.021], [9, 3834, 3.018]
2	[9, 133, 3.22], [10, 171, 2.956], [9, 232, 2.47], [10, 2342, 2.701], [11, 7746, 2.849], [10, 10,823, 3.432]
3	[10, 459, 3.062], [9, 1833, 2.394], [10, 108, 2.927], [11, 4336, 2.833]
4	[3, 2257, 1.103], [2, 28, 1.018], [1, 25,221, 0.985], [2, 14, 1.079], [3, 1371, 1.186], [2, 542, 1.081], [1, 54, 0.993], [2, 300, 1.092], [1, 1542, 0.978], [2, 186, 1.078], [3, 1183, 1.194], [2, 31, 1.098], [1, 5518, 1.033], [2, 10, 1.137], [3, 10,361, 1.202], [4, 8, 1.302], [5, 13, 1.392], [6, 14, 1.488], [7, 17, 1.598], [8, 198, 2.171], [9, 133, 2.315], [10, 1593, 2.769], [9, 25, 2.521], [10, 41, 2.76], [9, 18, 2.711], [10, 1962, 2.981]
5	[10, 2963, 2.692], [11, 12,012, 3.119], [10, 6118, 3.214]
6	[10, 2706, 2.531], [11, 337, 2.663], [10, 6630, 2.533], [9, 41, 2.321], [8, 31, 2.158], [7, 29, 2.081], [6, 50, 1.841], [5, 28, 1.629], [4, 34, 1.673], [3, 1225, 1.147], [4, 24, 1.207], [5, 21, 1.303], [4, 10, 1.209], [3, 2469, 1.138], [2, 12, 1.054], [1, 25,224, 1.236], [2, 12, 1.194], [3, 20,965, 1.246], [4, 13, 1.359], [5, 27, 1.45], [6, 29, 1.582], [7, 52, 1.71], [8, 1553, 2.137], [9, 1288, 2.808], [10, 13, 2.956], [9, 1497, 2.861], [10, 22, 3.053]
7	[10, 2864, 2.381], [11, 815, 2.514], [10, 2185, 2.327], [11, 10,150, 2.726], [10, 78, 2.646], [11, 1117, 3.062], [10, 539, 2.73], [11, 943, 3.104], [10, 1869, 3.005]
8	[14, 354, 3.261], [13, 1197, 3.135], [12, 1246, 3.235], [13, 76, 3.55], [14, 1561, 3.403], [13, 4936, 3.047], [12, 6, 3.058], [11, 17, 2.708], [10, 9, 2.717], [9, 8, 2.428], [8, 26, 2.028], [7, 7, 1.933], [6, 18, 1.758], [5, 15, 1.489], [4, 23, 1.468], [3, 18, 1.329], [2, 16, 1.211], [1, 49,629, 1.24], [2, 8, 1.354], [3, 18, 1.479], [4, 12, 1.586], [5, 10, 1.724], [6, 18, 1.815], [7, 14, 1.941], [8, 17, 2.081], [9, 21, 2.198], [10, 143, 2.287], [11, 199, 2.54], [12, 165, 3.064], [13, 2668, 2.987]
9	[13, 1207, 3.154], [12, 1554, 3.078], [11, 413, 2.981], [12, 345, 3.285], [13, 212, 3.532], [12, 287, 3.321], [11, 332, 3.259], [12, 1899, 3.295], [13, 16,379, 3.137]
10	[13, 1693, 3.552], [12, 606, 2.927], [11, 1570, 3.011], [12, 2772, 2.813], [11, 901, 2.937], [12, 1867, 3.106], [13, 272, 3.37], [12, 8617, 2.988], [11, 1867, 2.829], [12, 93, 3.053], [11, 242, 2.834]
11	[11, 1199, 3.016], [10, 1321, 2.79], [11, 50, 3.023], [12, 749, 3.126], [11, 1813, 2.931], [10, 157, 2.749], [11, 46, 2.978], [10, 89, 2.788], [11, 642, 3.023], [10, 1240, 2.713], [11, 144, 2.996], [12, 871, 3.116], [11, 1030, 2.973], [12, 434, 3.102], [11, 433, 2.776], [12, 2316, 3.037], [11, 730, 2.809], [12, 1011, 3.003], [11, 209, 2.718], [12, 803, 3.098], [11, 245, 2.838], [12, 1328, 3.067], [11, 1181, 2.968], [12, 476, 2.884], [11, 2, 2.88], [10, 22, 2.694], [11, 2433, 2.827]
12	[11, 2847, 2.997], [10, 926, 2.73], [9, 254, 2.358], [10, 237, 2.674], [11, 2804, 2.887], [12, 90, 2.884], [11, 5716, 2.827], [12, 2492, 2.997], [11, 154, 2.822], [12, 362, 2.828], [11, 1897, 2.777], [12, 31, 2.936], [13, 3342, 3.121]
13	[13, 9630, 3.09], [12, 4765, 3.047], [13, 893, 3.306], [12, 632, 3.043], [13, 96, 3.045], [12, 685, 2.967], [11, 160, 2.901], [12, 1619, 3.172], [13, 44, 3.38], [12, 1993, 3.009], [11, 101, 3.019]
14	[11, 874, 3.023], [10, 119, 2.686], [11, 402, 2.937], [10, 566, 2.608], [11, 1176, 2.835], [12, 337, 3.086], [11, 169, 2.827], [12, 85, 2.877], [11, 71, 2.834], [12, 2331, 3.168], [13, 172, 3.087], [12, 33, 3.27], [13, 4187, 3.257], [12, 2145, 3.142], [11, 30, 2.929], [12, 1333, 3.067], [11, 90, 2.862], [12, 151, 3.099], [11, 28, 2.862], [12, 2765, 3.106], [11, 488, 2.939], [12, 3010, 3.256], [11, 455, 2.869], [12, 76, 3.047]
15	[12, 99, 3.141], [11, 54, 2.988], [12, 704, 3.116], [11, 1777, 2.955], [12, 4382, 3.034], [11, 597, 2.902], [12, 39, 3.084], [11, 996, 2.879], [12, 594, 2.941], [11, 117, 2.844], [12, 1047, 3.11], [11, 492, 2.856], [12, 318, 3.204], [11, 1339, 2.966], [10, 173, 2.764], [11, 2584, 2.868], [12, 2096, 3.087], [11, 376, 2.867], [12, 87, 3.096], [13, 111, 3.281], [14, 3890, 3.624], [13, 2529, 3.21]
16	[13, 696, 3.315], [14, 2969, 3.595], [13, 227, 3.261], [14, 793, 3.271], [13, 1107, 3.287], [14, 1294, 3.598], [13, 201, 3.332], [14, 2881, 3.601], [13, 248, 3.309], [14, 4039, 3.469], [13, 218, 3.229], [14, 2552, 3.338], [13, 2035, 3.191]

Table 4 (continued)

No	STP
17	[13, 9219, 3.117], [12, 6, 2.932], [11, 11, 2.707], [10, 9, 2.51], [9, 7, 2.346], [8, 10, 2.184], [7, 7, 2.021], [6, 31, 1.915], [5, 35, 1.791], [4, 13, 1.649], [3, 11, 1.518], [2, 7, 1.396], [1, 49,802, 1.349], [2, 10, 1.472], [3, 12, 1.609], [4, 9, 1.726], [5, 9, 1.882], [6, 15, 2.174], [7, 14, 2.314], [8, 7, 2.297], [9, 28, 2.612], [10, 52, 2.583], [11, 155, 2.797], [12, 171, 2.935], [13, 1147, 3.029]
18	[13, 4962, 3.208], [14, 18,848, 3.23]
20	[14, 604, 3.354], [13, 1784, 3.353], [14, 7037, 3.367], [13, 266, 3.131], [12, 9, 2.89], [11, 10, 2.686], [10, 14, 2.512], [9, 7, 2.351], [8, 9, 2.199], [7, 5, 2.043], [6, 15, 1.893], [5, 20, 1.744], [4, 35, 1.601], [3, 13, 1.464], [2, 11, 1.337], [1, 30,884, 1.353], [2, 54, 1.617], [3, 19,184, 1.521], [4, 13, 1.793], [5, 9, 1.786], [6, 12, 1.916], [7, 15, 2.055], [8, 13, 2.216], [9, 17, 2.344], [10, 115, 2.587], [11, 154, 2.473], [12, 210, 3.025], [13, 2815, 3.025]
21	[13, 733, 3.265], [12, 996, 3.082], [11, 1153, 2.936], [12, 48, 3.317], [13, 2006, 3.403], [14, 26, 3.56], [13, 9097, 3.205], [14, 8924, 3.215]

Table 5 Input and results of the proposed algorithm

No	PN	Initial_Tap	WCT	Method	ET	TPQ	TEP	PPQ
1	Product A	11	4000	BC_Deviation()	148	9365.10	3745.12	2.50
2	Product A	11	6000	BC_Deviation()	201	15,360.97	6172.20	2.49
3	Product A	11	8000	BC_Deviation()	275	21,600.47	8588.68	2.51
4	Product A	11	10,000	BC_Deviation()	345	27,816.90	11,024.29	2.52
5	Product B	11	8000	BC_Average()	287	21,695.28	19,790.49	1.10
6	Product B	11	10,000	BC_Average()	365	27,620.33	22,185.41	1.24
7	Product C	11	8000	BC_MinMax()	546	3500.59	1263.89	2.77
8	Product C	11	10,000	BC_MinMax()	686	9932.36	3595.58	2.76

268 s, and then moves to tap position 10, where it stays for 10 s. The tap position movements continue until the final tap position 12 is achieved. The final tap stays for 1,086 s.

To further understand this, we explain the meaning of the effective tap sequence obtained through the proposed algorithm. Figure 10 shows two different tap working patterns between the original pattern and the pattern of the BC_MinMax() algorithm. The average PPQ value for the original tap working pattern is 6.57, while the PPQ value for the algorithm is 3.28. Moreover, the ET is 1,483 s.

Figure 11 reveals two different tap working patterns between the original pattern and the pattern of the BC_Average() algorithm. The average PPQ value for the original tap working pattern is 6.57, while the PPQ value for the algorithm is 3.31. Moreover, the ET is 1,842 s.

Figure 12 highlights two different tap working patterns between the original pattern and the pattern of the BC_Deviation() algorithm. The average PPQ value for the original tap working pattern is 6.57, while the PPQ value for the algorithm is 3.27. Moreover, the ET is 1,859 s.

Table 6 Tap sequence results obtained by the proposed algorithm

No	Tap sequence obtained by the proposed algorithm
1	[1, 5, 11, 12], [11, 61], [2, 12], [11, 76], [3, 12], [11, 84], [1-4, 6, 10-13, 15, 19], [11, 73], [2-4, 11, 12, 14, 29], [11, 40], [2, 3, 11, 12], [11, 39], [2-4, 11, 12], [11, 65], [2, 3, 11, 12, 21], [11, 42], [2, 3, 5, 10-12, 14], [11, 37], [1, 2, 7, 11, 12], [12, 41], [1, 10, 12, 23], [10, 54], [11, 67], [1-4, 11, 12, 17, 22], [11, 37], [1, 12], [11, 94], [1, 2, 4, 11, 12], [11, 117], [10, 556], [14, 549], [11, 359], [12, 34], [11, 262], [12, 337], [11, 169], [12, 85], [11, 71], [12, 201]
2	[1, 5, 11, 12], [11, 61], [2, 12], [11, 76], [3, 12], [11, 84], [1-4, 6, 10-13, 15, 19], [11, 73], [2-4, 11, 12, 14, 29], [11, 40], [2, 3, 11, 12], [11, 39], [2-4, 11, 12], [11, 65], [2, 3, 11, 12, 21], [11, 42], [2, 3, 5, 10-12, 14], [11, 37], [1, 2, 7, 11, 12], [12, 41], [1, 10, 12, 23], [10, 54], [11, 67], [1-4, 11, 12, 17, 22], [11, 37], [1, 12], [11, 94], [1, 2, 4, 11, 12], [11, 117], [10, 556], [14, 549], [11, 359], [12, 34], [11, 262], [12, 337], [11, 169], [12, 85], [11, 71], [12, 2201]
3	[1, 5, 11, 12], [11, 61], [2, 12], [11, 76], [3, 12], [11, 84], [1-4, 6, 10-13, 15, 19], [11, 73], [2-4, 11, 12, 14, 29], [11, 40], [2, 3, 11, 12], [11, 39], [2-4, 11, 12], [11, 65], [2, 3, 11, 12, 21], [11, 42], [2, 3, 5, 10-12, 14], [11, 37], [1, 2, 7, 11, 12], [12, 41], [1, 10, 12, 23], [10, 54], [11, 67], [1-4, 11, 12, 17, 22], [11, 37], [1, 12], [11, 94], [1, 2, 4, 11, 12], [11, 117], [10, 556], [14, 549], [11, 359], [12, 34], [11, 262], [12, 337], [11, 169], [12, 85], [11, 71], [12, 2331], [2, 13], [11, 170], [1, 12, 13, 33], [11, 447], [13, 102], [11, 503], [12, 612]
4	[1, 5, 11, 12], [11, 61], [2, 12], [11, 76], [3, 12], [11, 84], [1-4, 6, 10-13, 15, 19], [11, 73], [2-4, 11, 12, 14, 29], [11, 40], [2, 3, 11, 12], [11, 39], [2-4, 11, 12], [11, 65], [2, 3, 11, 12, 21], [11, 42], [2, 3, 5, 10-12, 14], [11, 37], [1, 2, 7, 11, 12], [12, 41], [1, 10, 12, 23], [10, 54], [11, 67], [1-4, 11, 12, 17, 22], [11, 37], [1, 12], [11, 94], [1, 2, 4, 11, 12], [11, 117], [10, 556], [14, 549], [11, 359], [12, 34], [11, 262], [12, 337], [11, 169], [12, 85], [11, 71], [12, 2331], [2, 13], [11, 170], [1, 12, 13, 33], [11, 447], [13, 102], [11, 503], [12, 2612]
5	[11, 291], [12, 1154], [11, 841], [12, 81], [6, 11, 12, 14], [11, 2086], [12, 3527]
6	[11, 291], [12, 1154], [11, 841], [12, 81], [6, 11, 12, 14], [11, 2086], [12, 5335], [13, 192]
7	[11, 268], [1, 4, 10, 12], [11, 78], [10, 3435], [11, 44], [10, 160], [11, 257], [10, 626], [1, 11, 13], [12, 1939], [10, 99], [1, 11], [12, 1086]
8	[11, 268], [1, 4, 10, 12], [11, 78], [10, 3435], [11, 44], [10, 160], [11, 257], [10, 626], [1, 11, 13], [12, 1939], [10, 99], [1, 11], [12, 3086]

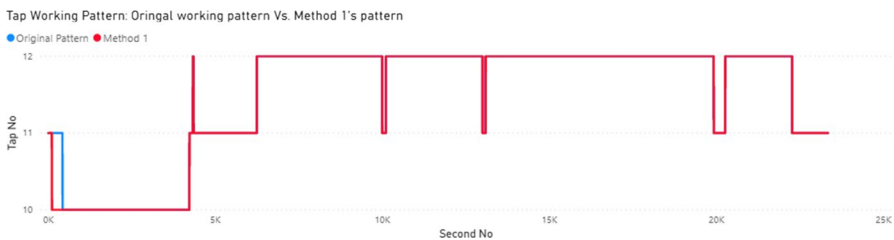


Fig. 10 Comparison of tap working patterns between the original pattern and BC_MinMax()’s pattern

The PPQ is obtained by applying the proposed dynamic programming technique to four products with different WCT and selection methods, as shown in Fig. 13. Similar to the original results, it was found that different products have different levels of PPQs, i.e., the lowest PPQ was obtained for product A, while the highest PPQ was obtained for product C. The PPQ results using the selection

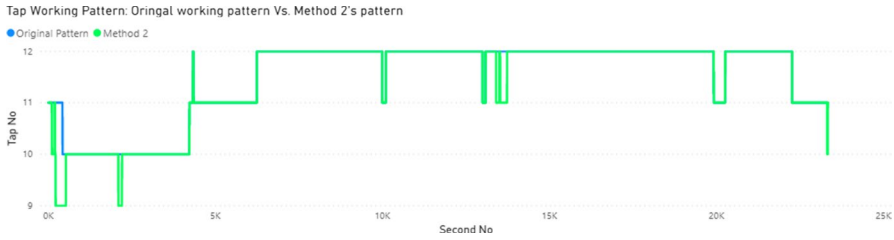


Fig. 11 Comparison of tap working patterns between the original pattern and BC_Average()'s pattern

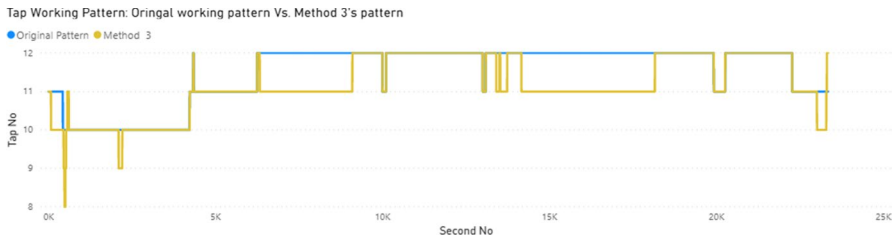


Fig. 12 Comparison of tap working patterns between the original pattern and BC_Deviation()'s pattern

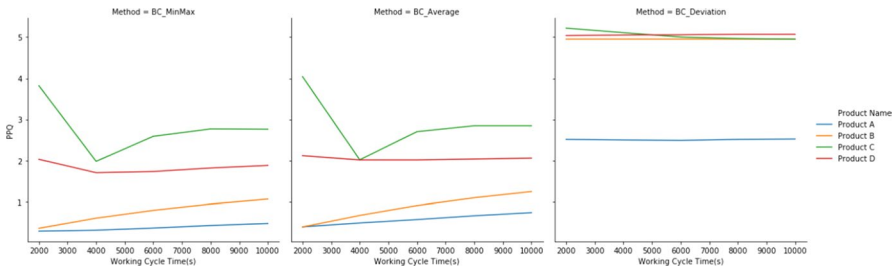


Fig. 13 Comparison of PPQs according to the products and selection methods

methods BC_MinMax() and BC_Average() exhibited similar patterns, whereas the BC_Deviation() method demonstrated different patterns for the PPQ.

5.4 Performance evaluation

To evaluate and measure the model, the mean absolute error (MAE) and root mean square error (RMSE) were used as the metric indicators to validate the model performance. One of the most commonly used performance metrics is the MAE, which enables the computation of errors. The continuous variables were measured to determine the error in the prediction of the model. It is well-known that the prediction error denotes the actual value minus the predicted value.

Table 7 Summary of performance metrics

Method	BC_MinMax	BC_Average	BC_Deviation
Number of data	82.164	82.164	82.164
RMSE	6.321	5.928	5.318
MAE	4.611	3.281	3.142

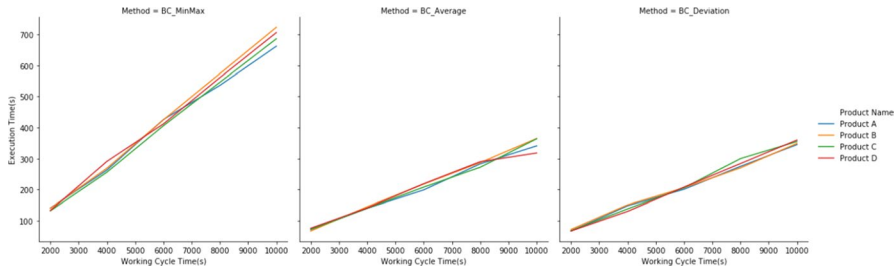


Fig. 14 Comparison of ETs according to the products and selection methods

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \tag{6}$$

where y_i is the real data, \hat{y}_i is the predicted data, and n denotes the number of observations in the model. The result of subtraction should be an absolute value. In this case, the smaller the error, the better the model.

Another performance measure is the RMSE; it denotes the performance of the model based on the result of prediction error.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \tag{7}$$

It should be noted that the MAE is different from the RMSE; the latter takes the square of residuals instead of similar weight. Consequently, the standard deviation of prediction with respect to errors has mutual correlation with the changes in the occurrence of distribution errors.

To compare the performances of these selection methods, a base case was created according to the effective result of PPQ with the same significant values, such as duration of sequence, initial tap position, and PN. Product A had a base case value of 4.82. A summary of the RMSE and MAE is presented in Table 7.

Finally, we present the ET of the proposed algorithm. As shown in Fig. 14, the ET for all cases increased linearly based on the WCT. These results indicate the theoretical time complexity of the dynamic programming algorithm based on the tap position movement constraint. (The tap only moves to the immediately adjacent position.) In other words, the results are attributable to the tap position at each second during the WCT, determined from the three adjacent tap positions.

6 Conclusions

In conclusion, this study contributes to the reduction in elemental power in an EAF environment by analyzing the working patterns. It should be noted that the power consumption burden can be solved by considering the tap position controls. In addition, this study presented evidence of the benefit of implementing a dynamic programming approach to determine the optimal value of PPQ with respect to the input parameters of tap position number and duration of the previous large amount of data, which stores the analysis result of tap position sequence and elemental power every second. However, the research has some limitations in terms of data correctness and data validation regarding the manual daily working pattern information. In future work, we plan to investigate the weighting factor and EPI to validate the use of power in EAFs. Furthermore, machine learning [31] and deep learning [32, 33] techniques will be implemented to obtain effective working tap positions in real time.

References

1. Hudson R, Sadler D (2017) The international steel industry: restructuring, state policies and localities. Routledge, London
2. Bradley F, and Nicholas W (2019) Worldsteel Short Range Outlook 2019, [Online]. Available: <http://www.worldsteel.org/media-centre/press-releases/2019/worldsteel-short-range-outlook-2019.html>. [Accessed: 12-Dec-2019].
3. Bradley F, and Nicholas W (2019) World Steel in Figures 2019 now available, [Online]. Available: <http://www.worldsteel.org/media-centre/press-releases/2019/world-steel-in-figures-2019.html>. [Accessed: 12-Dec-2019].
4. Madias J (2013) Electric arc furnace, Ironmaking and Steelmaking Processes: Greenhouse Emissions, Control, and Reduction
5. Kovačič M, Stopar K, Vertnik R, Šarler B (2019) Comprehensive electric arc furnace electric energy consumption modeling: a pilot study. *Energies* 12(11):2142. <https://doi.org/10.3390/en12112142>
6. Darmana D, Olsen JE, Tang K and Ringldalen E (2012) Modelling concept for submerged arc furnaces. In: The 9th International Conference on CFD in the Minerals and Process Industries CSIRO. Melbourne, Australia
7. Friedrich B, Kalisch M, Friedmann D, Degel R, Kaußen F, Bohlke J (2018) The submerged arc furnace (SAF): state-of-the-art metal recovery from nonferrous slags. *J Sustaine Metal*. <https://doi.org/10.1007/s40831-017-0153-1>
8. Saevarsdottir GA and Bakken JA (2010) Current distribution in submerged arc furnaces for silicon metal /ferrosilicon production. The 12th International Ferroalloys Congress, pp.717–728
9. Tesfahunegn YA, Magnusson T, Tangstad M and Saevarsdottir G (2018) Dynamic current distribution in the electrodes of submerged arc furnace using scalar and vector potentials, *Computational Science – ICCS 2018* pp 518–527
10. Minoru K, Masao M and Yoshiyuki K (1983) Improvement of the electric power consumption in silicomanganese smelting. *Proceedings of INFACON III, Tokyo, Japan*, 8–11
11. Kovacic M, Stopar K, Vertnik R, Šarler B (2019) Comprehensive electric arc furnace electric energy consumption modeling: a pilot study. *Energies* 12:2142. <https://doi.org/10.3390/en12112142>
12. Klemen S, Kovacic M, Peter K, Jože P (2014) Electric-arc-furnace productivity optimization. *Mater Tehnol* 48:3–7
13. Cano-Plata EA, Ustariz-Farfán AJ, Estrada JH (2018) Programming of operation in electric arc furnaces. *IEEE Trans Ind Appl* 54(4):3902–3908. <https://doi.org/10.1109/TIA.2018.2814978>

14. Mesa Fernández JM, Cabal VA, Montequin VR, Balsera JV (2018) Online estimation of electric arc furnace tap temperature by using fuzzy neural networks. *Eng Appl Artif Intell* 21:1001–1012
15. Gajic D, Savic-Gajic I, Savic I, Georgieva O, Di Gennaro S (2016) Modelling of electrical energy consumption in an electric arc furnace using artificial neural networks. *Energy* 108:132–139. <https://doi.org/10.1016/j.energy.2015.07.068>
16. Shyamal S, Swartz CLE (2019) Real-time energy management for electric arc furnace operation. *J Process Control* 74:50–62. <https://doi.org/10.1016/j.jprocont.2018.03.002>
17. Olczykowski Z (2003) Methods of determination of the voltage fluctuations and light flicker at simultaneous operation of three-phase arc furnaces. *Electr Power Qual Util* 9(1):47–58
18. Al-Zaid RZ, Al-Sugair FH, Al-Negheimish AI (1997) Investigation of potential uses of electric-arc furnace dust (EAFD) in concrete. *Cem Concr Res* 27(2):267–278
19. Rojas MF, De Rojas MS (2004) Chemical assessment of the electric arc furnace slag as construction material: expansive compounds. *Cem Concr Res* 34(10):1881–1888
20. King PE, Nyman MD (1996) Modeling and control of an electric arc furnace using a feedforward artificial neural network. *J Appl Phys* 80(3):1872–1877
21. Ma TLW, Sedighy M, Perkins BK, Gerritsen TA, and Rajda J (2003) Power control system for AC electric arc furnace
22. Trageser JJ (1980) Power usage and electrical circuit analysis for electric arc furnaces. *IEEE Trans Ind Appl* 2:277–284
23. Borovský T, Kijac J, Domovec M (2010) The slag composition influence on the dephosphorization and on the lifetime of electric arc furnace hearth refractory lining. *Acta Metall Slovaca* 16(3):165–171
24. Fisher D, DeLine R, Czerwinski M, Drucker S (2012) Interactions with big data analytics. *Interactions* 19(3):50–59
25. Denardo EV (2012) Dynamic programming: models and applications. Courier Corporation, USA
26. Bertsekas DP, Bertsekas DP, Bertsekas DP, Bertsekas DP (1995) Dynamic programming and optimal control, vol 1. Athena scientific Belmont, MA
27. Sprinthall RC, Fisk ST (1990) Basic statistical analysis. Prentice Hall Englewood Cliffs, NJ
28. Perkel JM (2018) Why Jupyter is data scientists' computational notebook of choice. *Nature* 563(7732):145–147
29. T. E. Oliphant, A guide to NumPy, vol. 1. Trelgol Publishing USA, 2006.
30. Pedregosa F et al (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
31. G. Fox et al. (2019) "Learning Everywhere: Pervasive machine learning for effective High-Performance computation," ArXiv Prepr. ArXiv190210810
32. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press, Cambridge
33. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Sokchomrern Ean¹ · Manas Bazarbaev¹ · Keon Myung Lee¹ · Aziz Nasridinov¹ · Kwan-Hee Yoo¹

Sokchomrern Ean
chomrern@cbnu.ac.kr

Manas Bazarbaev
manas@chungbuk.ac.kr

Keon Myung Lee
kmlee@cbnu.ac.kr

Aziz Nasridinov
aziz@cbnu.ac.kr

¹ Department of Computer Science, Chungbuk National University, Cheongju 28644, Korea