Cellular Automata Modeling of Decarburization of Metal Droplets in Basic Oxygen Steelmaking using Lopez

SUBMITTED BY:

Balogun Adebusola (101027260)

December 20, 2017

"This report was prepared for Professor Gabriel Wainer in partial fulfillment of the requirements for the SYSC 5104 course- Methodologies for Discrete Event Modelling and Simulation."

Cellular Automata Modeling of Decarburization of Metal Droplets in Basic Oxygen Steelmaking using Lopez

Adebusola Balogun
Department of Systems and Computer Engineering
Carleton University
1125 Colonel By Drive
Ottawa, ON. K1S-5B6 Canada
busolabalogun@cmail.carleton.ca

Abstract – The process of producing steel involves extraction of impurities from the raw metal like carbon, silicon and phosphorus. Decarburization reduces the amount of carbon in the metal. Diffusion, reaction and replenishment are procedures associated with the decarburization technique. The model has incorporated diffusion and decarburization reaction based on probabilities to study the evolution of the system using the Lopez simulator. The Newman neighborhood is deployed in this model. The implementation of the diffusion model in CD++ was the major focus for the Assignment II. However, this project considers both the diffusion and reaction models in the CD++ and Lopez simulator. The aim is to build an advanced Cell-DEVS model.

Keywords: *Decarburization, Diffusion, Reaction, Steelmaking, Lopez simulator.*

1. Introduction

The steel making process involves refining a hot metal by eliminating impurities like silicon, carbon and phosphorus, to a particular level by oxidizing the impurities and removing their oxides from the hot metal [1].

The paper [1] presented in the Assignment II basically describes the entire framework and procedures of the steelmaking technique. Therefore, they have employed the use of cellular automata in modelling this phenomena due to its level of complexity in the traditional modelling technique. The metal phase and the slag phase are two major phases encountered in the steelmaking process.

The following describes the procedural stages in which the slag and metal are processed;

- separation of FeO from slag and, carbon from metal to the reaction interface,
- detachment of FeO into Fe and O, and oxygen diffusion through the interface into the metal droplet.
- reaction of Oxygen with Carbon at the interface and,
- removal of the products [1].

The decarburization model basically implies removal of carbon from the metal, and it comprises of both the diffusion and decarburization reaction subroutines. These submodels are incorporated based on probabilities to study the evolution of the system in cellular automata.

The behaviour of each subroutine is carefully examined and deployed to build advanced Cell-DEVS models.

Subsequent sections describe further the developed models, their basic definitions and rules, results obtained from the simulations

and conclusions from the comparison between CD++ and Lopez simulator.

2. Background

In CD++, the presence of two planes (e.g. the diffusion plane and the direction plane) creates a 3D model of the system. However, the Lopez simulator enables declaration of state variables (in 2D), in order to store values in the cells and to use several ports to allocate additional values to neighbouring cells [2]. Therefore, in this case, each cell has a diffusion and direction state variables.

The referenced paper for this project is (Kundu, T. K. (2016, February). Cellular Automata Modeling of Decarburization of Metal Droplets in Basic Oxygen Steelmaking. In *IOP Conference Series: Materials Science and Engineering* (Vol. 115, No. 1, p. 012001). IOP Publishing).

The Newman neighbourhood is deployed in this model as shown in figure (1) below, where each element associated with the decarburization is distributed in the cell space.

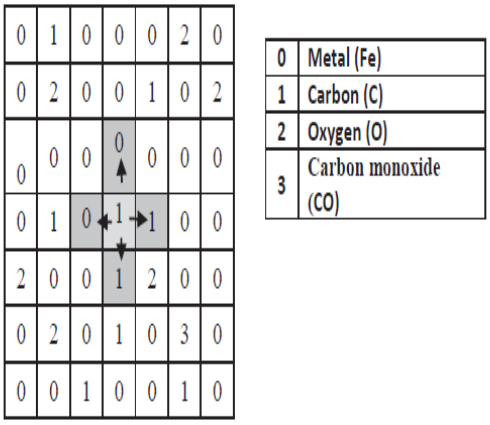

Figure (1). Newman Neighbourhood [1]

As depicted in the figure, each of the elements such as;

- Metal represents the state 0,
- Carbon represents the state 1,
- Oxygen represents the state 2, and
- Carbon monoxide represents the state 3 in the model.

The possible transition positions are shaded, and arrows are depicting a possible exchange during diffusion.

As discussed earlier, the decarburization process of steelmaking is divided into Diffusion, Reaction and Replenishment models. The subsequent subsections describe each one of these models and what has been achieved in the Assignment II.

2.1.Diffusion Model

This model is calculated by the random walk method; basically, spreading around the cell space.

The elements have the freedom to move in either of the cells in its Newman neighbourhood. There are two possible cases that are bound to happen in the diffusion model;

a) The elements can move to any one of the cells among the 4 shaded cells or,
b) The elements may decide not to move at all.

Only Carbon (C) and Oxygen (O) are the diffusing species, and the event for not moving is not considered for Oxygen.

The lattice considered consists of 20 x 20 square cells, and carbon is distributed in 18x18 grid leaving 2 rows of cells of Oxygen from boundaries and carbon (C), oxygen (O), carbon monoxide (CO) and iron (Fe) are the elements used.

For each Carbon cell, move to one cell in your neighborhood, depending on the random number chosen, which is a number between 10 & 14, if the random number is 10, remain in your current position.

Oxygen can move to one of its neighborhood cells depending on the random number chosen, but Oxygen cannot remain in its current position, it is always moving. Oxygen can go to one of the cells in its neighborhoods if CO does not block them. If Carbon diffuses by 1, Oxygen is allowed to diffuse by 100 steps.

In summary, if a Carbon cell will move to an Iron cell, Carbon & Iron will exchange positions. The state of cells between the movement occur also get exchanged. However, for one movement in Carbon, Oxygen moves 100 steps.

2.2.Reaction Model

The reaction model is based on the state of neighbouring cells. When Carbon (1) is in the same neighbourhood as Oxygen (2), reaction can occur and the state of the cell is changed to Carbon monoxide (3).

Although, the cell with state 2 which was involved in the reaction changes to Iron (0). If more than one cell with state 2 is present, one of them is chosen at random for the reaction. The reaction model is subdivided into two zones (Zone A and B) as shown in figure (2). Zone A is near the boundary and it is marked 5 cells from the boundary while the rest of the cells are in Zone B. Carbon was assumed to have reacted in Zone A, and the formed CO, from the reaction, has a greater chance of advancing towards halo formation as it has a higher chance of escaping into the boundary. On the other hand, the CO generated in Zone B has better chance of contributing to bloating of the droplet. The separation is completely random due to the unavailability of any qualified study and difficulty in conducting the experiments to quantify the rate of decarburization at various areas in a droplet [1].

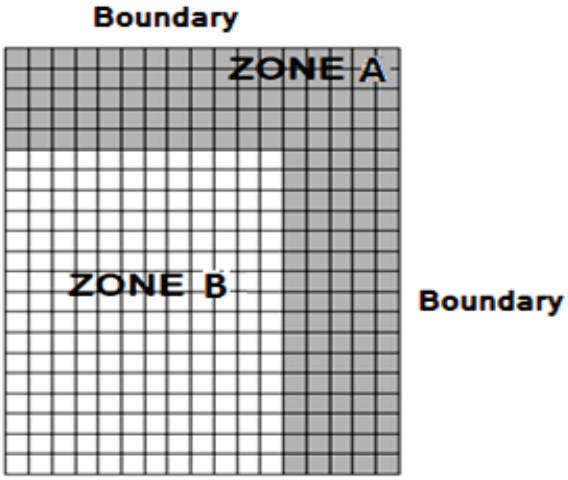

Figure (2). Reaction Zones [1]

2.3. Replenishment Model

The consumption of oxygen that took place in the reaction phase has given rise to the Replenishment model. The oxygen is resupplied to keep the concentration constant based on the equation below. Hence, the oxygen concentration in metal is determined by the FeO in slag.

$$\frac{[\%O]}{(\%FeO)} = (0.1T - 155.3) \times 10^{-4}$$

This given equation is used in the calculation of the mole fraction of Oxygen,

- where %FeO is the FeO concentration in the slag and,
- [%O] is the Oxygen concentration in the metal.
- T is the temperature range of the hot metal, which is between 1300-1400 degrees Celsius.

The initial distribution of C in the lattice is given based on its concentration as mole fraction.

Formal Cell-DEVS Specifications:

The formal specification for the Cell-DEVS Decarburization model is described as;

CD = < X, Y, I, S, θ, N, d, δint, δext, τ, λ, D >

X = Ø

Y = Ø

S = {0, 1, 2, 3}

N = neighbourhood = Von Neumann {(-1,0), (0,-1), (0,0), (0,1), (1,0)}

d = 100 ms

τ: N→S

3. Models Developed

The main focus of this project is on the implementation of the Diffusion and Reaction models using the Lopez simulator. In Assignment II, only the Diffusion model was developed, using CD++. The model was created in 3D, whereby there was a diffusion and direction plane. The figures below show results obtained in the simulation.

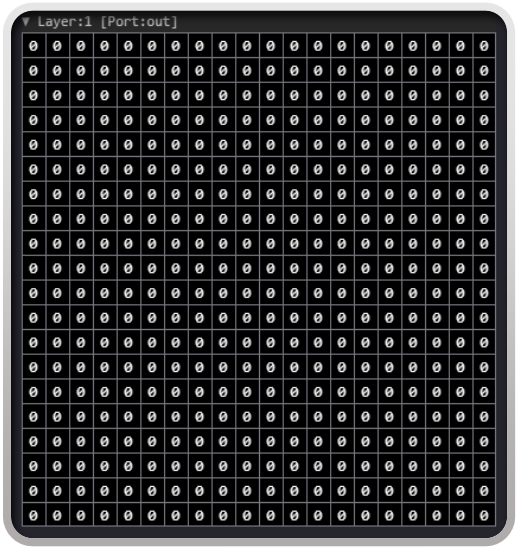

Figure (3). Initial phase of the Direction layer.

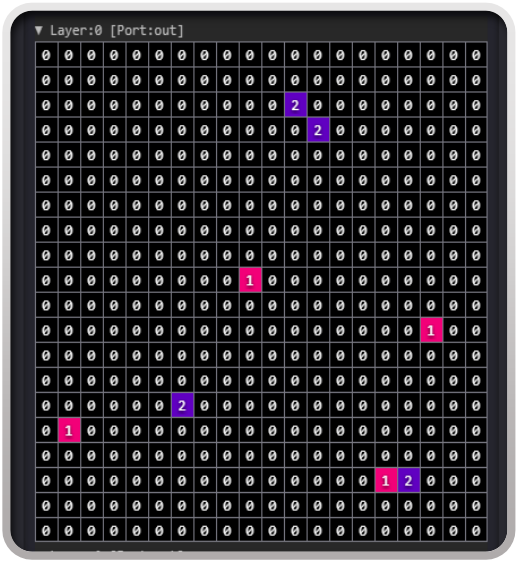

Figure (4). Initial phase of the Diffusion layer.

The final phase represents the state of the model when the diffusion process is active. The following figures show the final stage of diffusion between the cell states in their cell space.

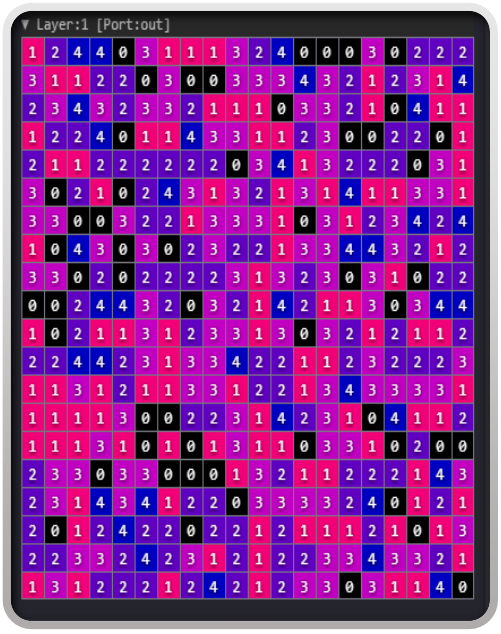

Figure (5). Final phase of the Direction layer.

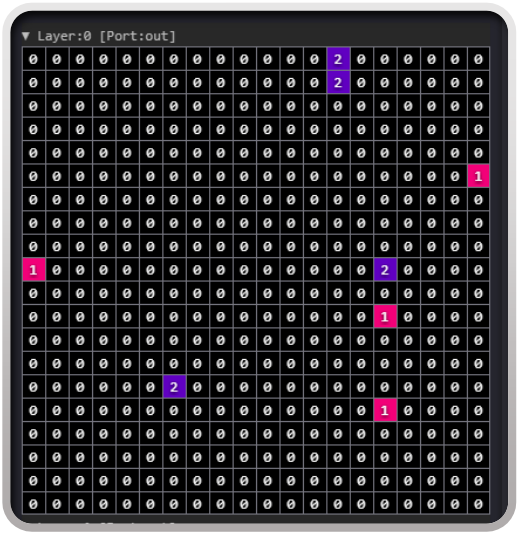

Figure (6). Final phase of the Diffusion Layer

3.1. Reaction Model in Lopez

Formal Cell-DEVS Specifications:

The formal specification for the Cell-DEVS Reaction model is defined by;

TDC = < *X*, *Y*, *I*, *S*, θ, *E*, delay, d, δint, δext, τ, λ, D > , where;

X = Ø

Y = Ø

S = {0, 1, 2, 3}

N = neighbourhood = Von Neumann {(-1,0), (0,-1), (0,0), (0,1), (1,0)}

d = 100 ms

τ: N→S:

S: //rules as shown in Reaction.ma.

[top]
components : Reaction

[Reaction]
type : cell
dim : (20,20)
delay : transport
defaultDelayTime : 100
border : nowrapped
%Von Newman neighborhood
neighbors : Reaction(-1,0) Reaction (0,-1) Reaction(0,0) Reaction(0,1)
neighbors : Reaction(1,0)
initialvalue : 0
initialvariablesvalue : reaction.stvalues
%Initial value file
localtransition : Reaction-rule

The 'value' state variable can have values 0 to 3 where Iron = 0, Carbon = 1, Oxygen = 2, Carbon Monoxide (CO) = 3.
This variable holds the value or state of each cell.
statevariables : value
% Port for the value state variable
neighborports : valueport
stateValues : 0

[Reaction-rule]
If a cell with state 1 (i.e. carbon) has a cell with state 2 (i.e. oxygen) in its Newman neighborhood, then the reaction can occur and the state of the cell is changed to 3 (i.e. carbon monoxide or CO).
There are four possible cases associated with the Reaction model, which each case has rules.

Case 1: Carbon has only 1 oxygen cell in its neighborhood.

- Oxygen is in the North;

rule : {~valueport := $value;} {$value := 11;} 100 { (0,0)~valueport = 1 AND #macro(North) = 2 AND stateCount(2,~valueport) = 1 }

- Oxygen is in the East;

rule : {~valueport := $value;} {$value := 12;} 100 { (0,0)~valueport = 1 AND #macro(East) = 2 AND stateCount(2,~valueport) = 1}

- Oxygen is in the South;

rule : {~valueport := $value;} {$value := 13;} 100 { (0,0)~valueport = 1 AND #macro(South) = 2 AND stateCount(2,~valueport) = 1}

- Oxygen is in the West;

rule : {~valueport := $value;} {$value := 14;} 100 { (0,0)~valueport = 1 AND #macro(West) = 2 AND stateCount(2,~valueport) = 1}

If more than one cell with state 2 is present, one of them is randomly selected for the reaction.

Case 2: Carbon has 2 oxygen cells in its neighborhood (randomly select one Oxygen cell).

- Oxygen is in the North and East (NE);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, 11,12);} 100 { (0,0)~valueport = 1 AND #macro(North) = 2 AND #macro(East) = 2 AND stateCount(2,~valueport) = 2}

- Oxygen is in the North and South (NS);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, 11,13);} 100 { (0,0)~valueport = 1 AND #macro(North) = 2 AND #macro(South) = 2 AND stateCount(2,~valueport) = 2}

- Oxygen is in the North and West (NW);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, 11,14);} 100 { (0,0)~valueport = 1 AND #macro(North) = 2 AND #macro(West) = 2 AND stateCount(2,~valueport) = 2}

- Oxygen is in the South and East (SE);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, 13,12);} 100 { (0,0)~valueport = 1 AND #macro(South) = 2 AND #macro(East) = 2 AND stateCount(2,~valueport) = 2}

- Oxygen is in the South and West (SW);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, 13,14);} 100 { (0,0)~valueport = 1 AND #macro(South) = 2 AND #macro(West) = 2 AND stateCount(2,~valueport) = 2}

- Oxygen is in the East and West (EW);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, 12,14);} 100 { (0,0)~valueport = 1 AND #macro(East) = 2 AND #macro(West) = 2 AND stateCount(2,~valueport) = 2}

Case 3: Carbon has 3 oxygen cells in its neighborhood (randomly select one Oxygen cell).

- Oxygen is in the North, South and East (NSE);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, if(randInt(1) = 0, 11,13), 12);} 100 { (0,0)~valueport = 1 AND #macro(North) = 2 AND #macro(South) = 2 AND #macro(East) = 2 AND stateCount(2,~valueport) = 3}

- Oxygen is in the North, South and West (NSW);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, if(randInt(1) = 0, 11,13), 14);} 100 { (0,0)~valueport = 1 AND #macro(North) = 2 AND #macro(South) = 2 AND #macro(West) = 2 AND stateCount(2,~valueport) = 3}

- Oxygen is in the North, East and West (NEW);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, if(randInt(1) = 0, 11,12), 14);} 100 { (0,0)~valueport = 1 AND #macro(North) = 2 AND #macro(East) = 2 AND #macro(West) = 2 AND stateCount(2,~valueport) = 3}

- Oxygen is in the South, East and West (SEW);

rule : {~valueport := $value;} {$value := if(randInt(1) = 1, if(randInt(1) = 0, 13,12), 14);} 100 { (0,0)~valueport = 1 AND #macro(South) = 2 AND #macro(East) = 2 AND #macro(West) = 2 AND stateCount(2,~valueport) = 3}

Case 4: Carbon has 4 oxygen cells in its neighborhood (randomly select one Oxygen cell).

- Oxygen is in the North, South, East and West (NSEW);

rule : {~valueport := $value;} {$value := round(uniform(11,14));} 100 { (0,0)~valueport = 1 AND #macro(North) = 2 AND #macro(South) = 2 AND #macro(East) = 2 AND #macro(West) = 2 AND stateCount(2,~valueport) = 4}

- Change the reacted oxygen cells to 0 (i.e. Fe);

rule : {~valueport := $value;} {$value := 0;} 0 { #macro(South) = 11 }
rule : {~valueport := $value;} {$value := 0;} 0 { #macro(West) = 12 }
rule : {~valueport := $value;} {$value := 0;} 0 { #macro(North) = 13 }
rule : {~valueport := $value;} {$value := 0;} 0 { #macro(East) = 14 }

- Change the reacted carbon cells to 3 (i.e. CO);

rule : {~valueport := $value;} {$value := 3;} 0 { (0,0)~valueport = 11 }
rule : {~valueport := $value;} {$value := 3;} 0 { (0,0)~valueport = 12 }
rule : {~valueport := $value;} {$value := 3;} 0 { (0,0)~valueport = 13 }
rule : {~valueport := $value;} {$value := 3;} 0 { (0,0)~valueport = 14 }
rule : {~valueport := $value;} {$value := (0,0);} 0 {t}

4. Simulation results

The reaction model was successfully implemented in CD++. Although, the Lopez simulator did not produce results when the created model was simulated on the software. The following figures represent the results for the Reaction model simulated in CD++.

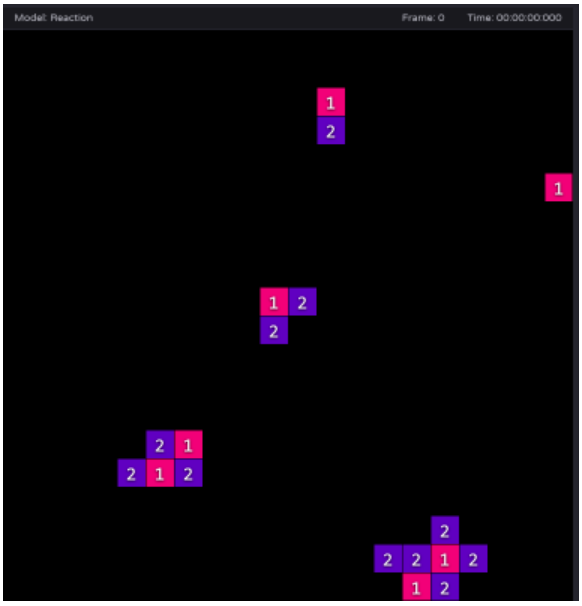

Figure (7). Initial phase of the Reaction model.

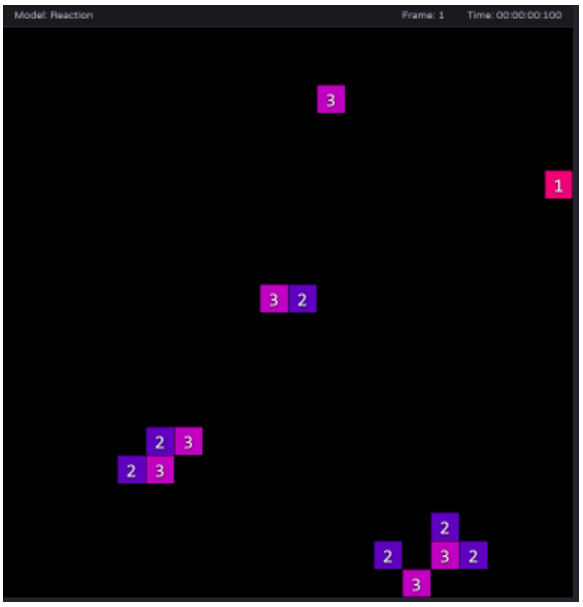

Figure (8). Final phase of the reaction model

5. Conclusion

The execution time of the Lopez simulator as compared to the older CD++ version, was found to be faster, and the simulation time was reduced.

Extension of the assignment II was majorly on implementing the reaction model in Lopez using state variables.

References

[1] Kundu, T. K. (2016, February). Cellular Automata Modeling of Decarburization of Metal Droplets in Basic Oxygen Steelmaking. In *IOP Conference Series: Materials Science and Engineering* (Vol. 115, No. 1, p. 012001). IOP Publishing.

[2] Model, C. D. M. D. D. (2003). Extending Extending CD++ Specification Language for Cell-DEVS Model Definition DEVS Model Definition DEVS Model Definition.