A Simulation Model For Equipment Selection In A Steel Plant

Pedro A. Nassau
Hylsa, S.A.
Monterrey, Mexico

ABSTRACT

A simulation model was developed to determine the optimal policy for utilization and replacement of moving equipment in a steel plant. Specifically, this paper deals with the problem of slag removal in a melt-shop composed of a batch of five electric-arc furnaces. The environment of high temperatures in which service has to take place leads to conditions of frequent equipment break-down and a dynamic rescheduling of activities, which makes the use of simulation a necessity. One important feature that was included in the model was the process of "aging", that is, that service time is a function of the number of minutes that the slag has been waiting for its removal.

INTRODUCTION

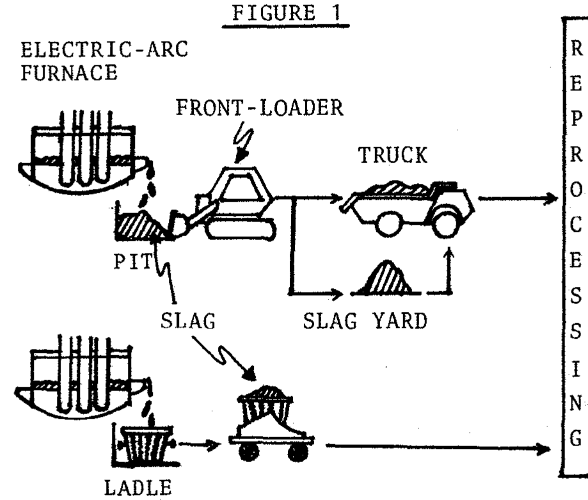
BACKGROUND

HYLSA (Monterrey Works) is an integrated steel producer, manufacturing a variety of flat-steel products. The current production is in the order of one million ingot tons a year, although studies for a major expansion are under way. The production center covered in this paper is the melt-shop, where raw materials (sponge iron and ferrous scrap) are transformed, by means of electric-arc furnaces, into steel ingots. During the melting and refining stages a layer of slag starts to form on the surface of the melting steel (similar to foam in a mug of beer) containing the impurities to be eliminated. On certain intervals the furnace is tilted to allow the slag to pour down, either into receiving ladles or directly on the ground. The final removal is obtained by two different (concurrent) means: first, if ladles are used, these are placed on specially designed railroad cars; a locomotive is then called to take the ladles to the reprocessing plant. Second, if ladles are <u>not</u> used, then the slag must first be removed from the "pit" beneath the furnaces (using front-loaders) and then, either loaded on a waiting truck or else accumulated in the slag-yard until such a unit becomes available. Finally, the truck, containing a full load of slag is dispatched to the reprocessing plant. The entire process is depicted in Figure 1.

NATURE OF PROBLEM

The slag-removal operation had been plagued with problems for many years. An insufficient number of service units, in addition to a lack of standard practices

FIGURE 1



had led to the following:

1. An extremely short economic life of the front-loaders.
2. A very low available time for these units.
3. High equipment maintenance costs.
4. High rental costs for additional units.

However, all of these amounted to very little compared to the occasional (but costly) shut-downs forced on the furnaces. This occured when the dripping slag piled up to a height that made the necessary pivoting motions of the furnaces impossible. The final decision to perform a systematic study was made when it was announced that the furnaces were being "charged-up" to increase their productivity by about 15% (achievable by a reduction in total cycle time). This would have placed an intolerable burden on the slag-handling facility.

THE MODEL

Several approaches were initially tried. One of these was to consider the system a machine-repair model of queueing theory. In it, a finite population (the furnaces) call upon the repairmen (the front-loaders) for service (clean-up). However, the underlying simplifying assumptions did not render a realistic representation of the actual situation; simulation was therefore adopted.

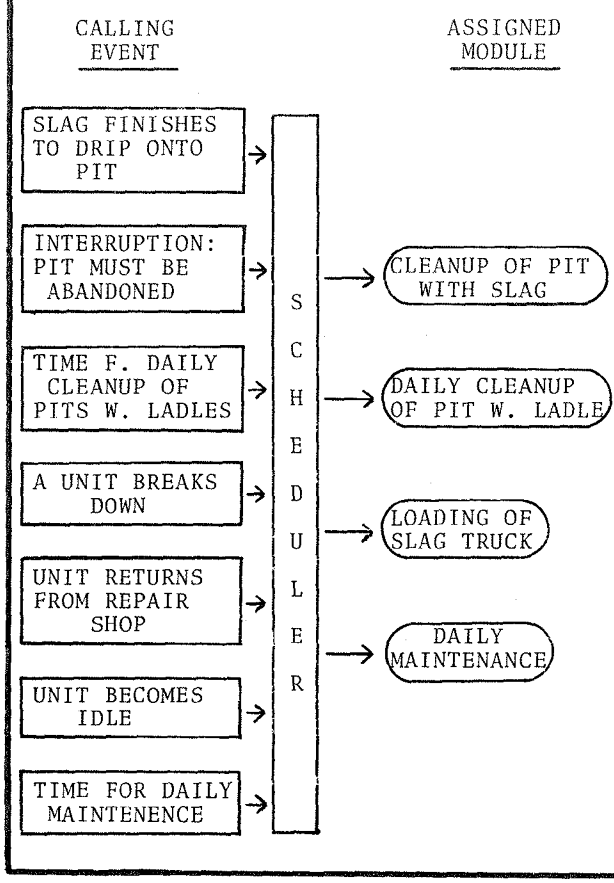
The model, built in GPSS, consists of two segments, which are basically independent of each other. The first one is rather straightforward and deals with the furnaces serviced by slag-ladles; the other one depicts the operation of front-loaders. This paper deals primarilly with the latter.

FRONT-LOADER SEGMENT

The assignment and control of front-loaders is governed by a central SCHEDULER module (Figure 2). Some of the tasks it performs are the following:

1. Keep track of the level of slag in each pit.
2. Identify the furnace(s) which has (have) reached a critical level of slag.
3. Record the number of available (working, cooling down and idle), and unavailable service units.

FIGURE 2
SCHEDULER MODULE



4. Forecast the next call for slag removal.

According to the changing conditions of the system, it identifies the task with the highest priority (following preestablished rules) and assigns an available front-loader to perform it. As Figure 2 shows, execution of the SCHEDULER is triggered by two kinds of events: a call for service or the return of an available unit.

The single most important operational segment is the CLEAN-UP OF PIT WITH SLAG. The main events that take place are illustrated in Figure 3. The segment is enriched, however, by a number of considerations that have to be taken into account, namely:

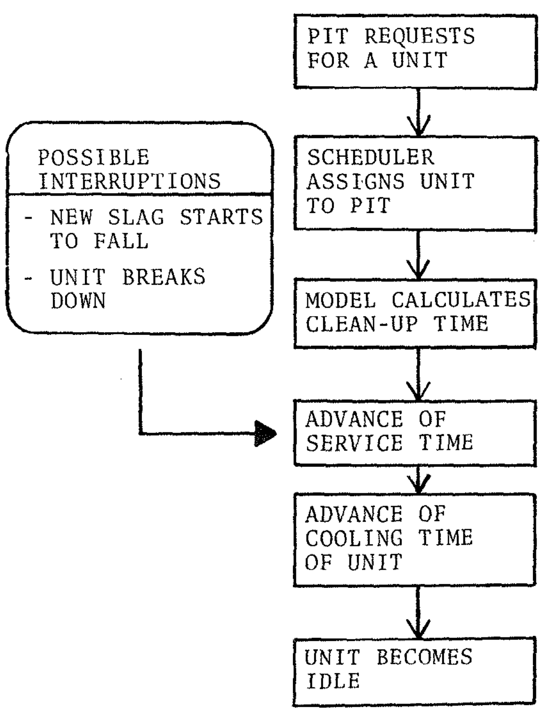
- The time period in which a front-loader may enter a pit to remove slag is limited. Once a furnace is roughly halfway through the melting stage, slag starts to drip onto the pile below, at which point the pit must be vacated. In practice, this means that the pit is out-of-bounds approximately 50% of the time.

- Break-down of service units occur at random. Therefore, at times, a front-loader is forced to leave a pit before it completes clean-up, whereby a new unit must be reassigned to finish the task.

- Due to the high temperatures surrounding the equipment, service units are not permitted to remain inside a slag-pit for more than 45 minutes at a time. Since this may be insufficient, the job is left incomplete.

- The slag removal time is not only a function of the actual amount piled up, but also of the "age" of the slag: the longer the slag has to wait for the equipment to arrive, the harder it gets and the longer it takes to being removed. For example, it takes about 2.5 minutes for the unit to remove one shovel (about 1.5 tons) and return to the pit. However, the same operation takes twice as long if the pit has been left unattended for more than three hours.

FIGURE 3
CLEAN-UP OF PIT WITH SLAG



Once the model was validated with historic data, a series of simulation runs was performed, parametrically changing each one of the following:

- Total number and capacity of front-loaders in the system.

- Availability of the service units.

- Total cycle time/furnace.

- Number of front-loaders serviced by front-loaders vs. serviced by slag-ladles.

- Number of equipment operators.

The methodology that was followed was to find the minimum availability of the front loaders that would assure an adequate level of service, while holding the remaining input data constant. A search procedure was devised for this purpose. The results were then presented in graphical form, a sample of which appears in Figure 4. In addition, each run was complemented with the following statistics:

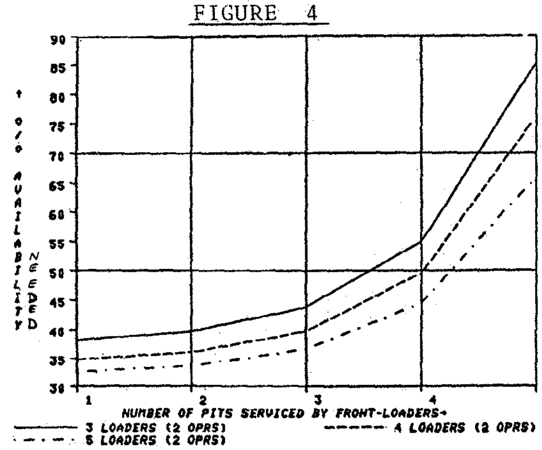
- average number of tons of slag/pit (steady state)

- the number of times a furnace had to be stopped because of excessive slag.

- equipment utilization (during available and total time).

- utilization of overhead crane in the ladle segment.

FIGURE 4



Based on these, the company's management was able to make the following decisions:

1. Immediate purchase of two new front-loaders, and hiring of one new operator per shift.

2. Elimination of the ladle system for the removal of slag.

3. A schedule for the replacement of units that fall below a reference percentage availability.

4. A realistic estimate of the number of units that will be needed in the future.

CONCLUSIONS

A number of insights were gained by the development of a simulation model. These have led to a better understanding of the slag-removal operation and have aided in the establishment of better control in the work area. The following are two of these:

1. A supervisor was made responsible for the dispatching of the available front-loaders and the repair of the damaged units. Previously, the operators had taken these tasks upon themselves, with little concern for the state of the machines.

2. The simulation model proved that although the units would be idle most of the time, they could not be reassigned to other tasks. The reason for this is the need for the units to be available when needed.

BIBLIOGRAPHY

1. Bobillier, P.A., Simulation with GPSS and GPSS V, Prentice-Hall Inc., Englewood Cliffs, N.J., 1976

FROM THE EDITOR'S DESK

(continued from page 21)

Fortran, Cobol, Pascal, Ada and PFFT!

Seldom do we have such fast turnaround time in SIMULETTER, but I felt that I had to add my two cents to Joe Clema's comments about ADA (see Chairman's Column on page 2 of this issue).

Let me start by saying that I hold no brief for ADA or for that matter PASCAL. However, after more than 20 years in computing, I find it both amusing and sad when a new language creates such violent reaction. I am sure that there were many of our members around when IBM introduced PL/I. I recall the howls of both the COBOL and FORTRAN camps. And it was only natural for many to join one side or the other against PL/I since it was also fashionable to oppose the Great White Father from Poughkeepsie. Well, PL/I came and to some extent went. Yet, PL/I is still alive and living not only in some universities but also in industry. At my college, for example, we continue to teach PL/I since many of our students, who continue their education at other institutions in our area on Long Island and even upstate, are required to have a knowledge of this language for several advanced courses in computing.

Possibly I should add that I learned 'Computing' by first studying machine language. After a brief switch to FORTRAN, I went to the Institute of Computer Science at the University of Toronto where C. Gotlieb introduced me and others on a Ford Foundation grant to the wonder of a yet to be born language, APL, by providing us with some background in 'Iverson notation.' I have therefore always felt that my 'mother tongue' in computing was Iverson, but because of compiler limitation at the College, I adopted FORTRAN. With FORTRAN 77, I feel that I can write any program, especially those developed by many associates who promote COBOL as the only language anyone should use.

Maybe it is this constant pressure from the COBOL adherents that makes me look to ADA as a way out. Even our students learn to walk like apes, carrying their hugh COBOL programs around. Dan McCracken, at a meeting I attended last year, repeated that old one about how much industry has invested in COBOL programs. This I can recognize and

(continued on page 54)