Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

SIMULATION OPTIMIZATION WITH THE LINEAR MOVE AND EXCHANGE MOVE OPTIMIZATION ALGORITHM

Marcos Ribeiro Pereira Barretto
Leonardo Chwif

Mechatronics Lab
University of São Paulo
Av. Prof. Mello Moraes 2231
Sao Paulo, 05508-900, BRAZIL

Tillal Eldabi
Ray J. Paul

Centre For Applied Simulation Modelling
Department Of Information Systems And Computing
Brunel University
Uxbridge, Middlesex, UB8 3PH, U.K.

ABSTRACT

The Linear Move and Exchange Move Optimization (LEO) is an algorithm based on a simulated annealing algorithm (SA), a relatively recent algorithm for solving hard combinatorial optimization problems. The LEO algorithm was successfully applied to a facility layout problem, a scheduling problem and a line balancing problem. In this paper we will try to apply the LEO algorithm to the problem of optimizing a manufacturing simulation model, based on a Steelworks Plant. This paper also demonstrates the effectiveness and versatility of this algorithm. We compare the search effort of this algorithm with a Genetic Algorithm (GA) implementation of the same problem.

1 INTRODUCTION

There are a number of characteristics that make simulation a powerful tool, considered as one of the most used techniques in OR. Firstly, its inherent ability to evaluate complex systems with a large number of variables and interactions for which there is no analytic solution. Secondly, simulation can model the dynamic and stochastic aspects of systems, generating more precise results when compared with static and deterministic spreadsheet calculations. It is also considered as a tool to answer “What if” questions. Simulation itself is a solution evaluator technique, not a solution generator technique (Harrell and Tumay 1994). This scenario could be changed with the aid of optimization procedures. In this case, simulation could answer not only “What if” questions but also answers “How to” questions (Azadivar 1992), providing with the best set of input variables that maximize or minimize some performance measure(s) (based on the modeling objectives).

According to Stuckman et al. (1991), simulationists can be classified into 3 categories with regards to the optimization of quantitative variables: the first category tend to use the "trial and error" method, varying the input variables in order to find which set gives the best performance. The second category tends to systematically vary the input variables, to see their effects on the output variables. The third category will apply an automated simulation optimization approach. In the paper we are concentrating on the last category. That is, an automated optimization approach.

The rest of this paper is organized as follows. In section 2 we present a brief review of simulation optimization. In Section 3, we describe LEO algorithm that is based on simulated annealing, developed to solve hard combinatorial optimization problems. In Section 4 we present the simulation model adopted for this study. In section 5 we report on the implementation of the LEO algorithm to the Steelworks simulation problem and show its results. Finally, section 6 draws conclusions.

2 BRIEF REVIEW OF A SIMULATION OPTIMIZATION PROBLEM

A simulation optimization problem could simply be defined as an optimization problem where the objective function, constraints or both are responses that can only be evaluated by computer simulation (Azadivar 1992). In these cases suppose a simulation model M, has *n* input variables ($x_1, x_2,\ldots x_n$) and *m* output variables ($y_1, y_2, \ldots y_m$). The objective of the Simulation Optimization is to find the optimum values ($x_1^*, x_2^* \ldots x_n^*$) for the input variables ($x_1, x_2,\ldots x_n$) that minimizes or maximizes the output variable(s). In order to solve a simulation optimization problem both the operational simulation model (i.e. simulation model implemented on the computer) and an optimization method or procedure are needed. Figure 1 depicts the interrelationship between the simulation model and the optimization procedure, where output from the simulation model is the input to the optimization procedure.

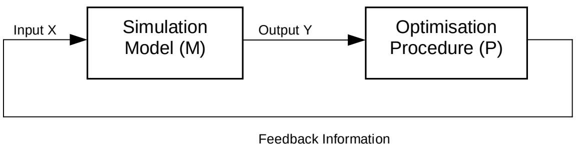

Figure 1: Relationship Between Simulation Model (M) And Optimization Procedure (P)

We can classify simulation optimization problems regarding their input variables (1), output variables (2), or by their Optimization Procedure (3). Regarding 1, we can have both quantitative variables and qualitative variables, regarding 2 we can have a single objective simulation optimization problems (if there is only one output variable) or multi-objective problems (for optimizing multiple output variables). The most reported optimization methods or procedures reported in the literature could be classified basically into four categories: Gradient based search, Stochastic Approximations, Response Surface Methodology and Heuristic Search Methods. The last one has been widely applied because of its ability to solve hard combinatorial optimization problems (including simulation optimization).

There are many articles in the literature regarding most known heuristic search methods. Here we give some: Tabu Search (TS) (Glover 1990); Genetic Algorithm (GA) (Goldberg 1989); and Simulated Annealing (SA) (Kirkpatrick 1983). GA application to simulation optimization problems is reviewed by Tompkins and Azadivar (1995). A comparison of SA and GA was made by Stuckman et al. (1991), with respects to simulation optimization problems. Both Lee and Iawate (1991) and Manz et al. (1989) presented an application to a simulation optimization problem using a SA algorithm.

Meketon (1987), Azadivar (1992), Fu (1994) and Carson and Maria (1997) survey this issue. The last makes also a review of some simulation optimization applications and present some commercial software available.

3 OPTIMIZATION PROCEDURE

3.1 Simulated Annealing

Since the LEO algorithm is based on a Simulated annealing algorithm, a brief explanation of this is given in this section. Simulated Annealing is a method based on Monte Carlo simulation, which solves difficult combinatorial optimization problems. The name comes from the analogy to the behavior of physical systems by melting a substance and lowering its temperature slowly until it reaches freezing point (physical annealing). Simulated annealing was first used for optimization by Kirkpatrick et al. (1983).

Suppose that there is a solution space *S* (the set of all solutions) and an objective function (*C*) (real function defined on members of *S*). The purpose is to find a solution (or state), $i \in S$ that minimizes *C* over *S*. SA makes use of an iterative improvement procedure which is determined by a neighborhood generation. So starting with an initial state, a neighbor state is generated, and the algorithm accepts or rejects this based on a certain criterion. In most cases, this acceptance is decided stochastically with the probability of $e^{-\delta/T}$ (Metropolis criterion), where δ is the difference in costs between the current state and its neighbor, and *T* is the current temperature (control parameter). This is a mechanism for avoiding being trapped in local optima. Note that higher the temperature, the higher will be the probability of uphill moves (moves that "worsen" the objective function).

Some choices must be made for any implementation of SA, which determines the cooling schedule. This consists in finding basically the initial value of temperature T_O, the temperature function which determines how the temperature is lowered, the number of iterations *N* (Epoch length) to be performed at each temperature or each "epoch" and the stopping criterion to terminate the method. A geometric temperature function is generally used, that is. $T*(t+1) = R*T(t)$ where *R* is a constant called the cooling ratio ($0<R<1$). *N* is also a constant called "Epoch Length" which represents the number of interactions made at each temperature level. The algorithm proceeds until the temperature reaches the final temperature T_f, which corresponds in the analogy, to the frozen temperature. Johnson (1989) has pointed out that the temperature value has a direct relationship with the neighborhood acceptance rate. Hence the initial and final temperature may be chosen by establishing their respective acceptance rates *AR* (T_O) and *AR* (T_f) (Johnson 1989; Van Laarhoven and Aarts 1987; Eglese 1990).

The general simulated annealing algorithm in pseudo code, with the adopted cooling schedule can be seen next:

```
Generate_state_i; {initial state}
T=To;
Repeat
    Repeat
      k:=0;
        Generate_state_j; {neighbour state}
        δ=C(j) - C(i);
        If δ<0 then i:=j
               else if random(0,1) < e^(-δ/T) then
                          i:=j;
        k:=k+1;
    until k=N;
    T:=R.T;
until T≤Tf.
```

For the above case *C* is a real number but also a deterministic objective function. Since we are dealing with a stochastic objective function *C* in the case of simulation and we have to generate multiple replications of the simulation model, the code in bold above has to be changed to:

δ=m(C(j)) –m(C(i));
If compare(m(C(j)),sd(C(j));
m(C(i)),sd(C(i)) then i:=j
else if random(0,1) < $e^{(-\delta/T)}$ then i:=j;

where *m(C(x))* is the mean of the objective function for state *x* and *sd(C(x))* is its standard deviation. In our case the comparison to know which objective function is better given a certain level of confidence follows the Welch Criteria (Law and Kelton 1991).

3.2 Linear And Exchange Move Optimization Algorithm (LEO)

The LEO algorithm uses Simulated Annealing with a specific neighborhood generation. With this approach, it is possible to solve combinatorial optimization problems with either real or integer variables. The LEO approach showed promising efficiency to solve layout problems (Chwif et al. 1998) and scheduling problems (Barretto et al. 1998).

Let's define *n* input variables ($x_1,x_2,\ldots.x_n$), real or integer, the set ($max_1,max_2,\ldots.max_n$) as their maximum values, the set ($mix_1,mix_2,\ldots.mix_n$) as the their minimum values and the set ($stepx_1,stepx_2,\ldots.stepx_n$) as their resolution. The LEO algorithm works with basically two procedures for neighborhood generation LEO. The first procedure increases or decreases the variable values according to their respective steps. The second changes its value for another value generated by a uniform distribution between mix and max rounded to their respective resolution. These two procedures are shown in pseudo code below:

LINEAR MOVE:

Choose RANDOMLY $1\leq k\leq n$
repeat
    case (direction) *of*
        "up": x_k=x_k-$stepx_k$;
        "down": x_k=x_k-$stepx_k$
until ($x_k\leq max_k$) *and* ($mix_k\leq x_k$)

EXCHANGE MOVE:

Choose RANDOMLY $1\leq k\leq n$
x_k=roundstep(uniform((mix_k, max_k))

In the first case (Linear Move) the values of the input changes "smoothly" by increasing or decreasing the input variable by one step. The probability of increasing or decreasing one variable is 50%. The choice of the variable to be modified on each neighborhood generation is random. In the second procedure (Exchange Move), after randomly choosing one of the input variables, it receives a completely new value within the interval given by [min_k, max_k] rounded according to its resolution. The second procedure makes a more sharp change on the variables, allowing the optimization algorithm to escape for instance from a region where the values of input variables give very bad results in terms of the objective function. The choice of which procedure to apply to each neighborhood generation (since for each neighborhood only one procedure has to be applied) is taken with a fixed probability of 50%.

4 CASE STUDY

4.1 Steelworks Model

The steelworks simulation model is an example of a manufacturing simulation model fully described in Paul and Balmer (1993). We will only provide a brief explanation here. In the Steelworks plant, there are two blast furnaces, which melt iron at certain daily volumes, which blows and fills as many torpedoes as available and are used to transport molten iron. If no torpedo is available, the molten iron is dropped on the floor and waste is produced. Each torpedo can hold a fixed quantity of molten iron. All torpedoes with molten iron go to a pit, where crane(s)-carrying ladles are filled from torpedoes, one at a time. The ladle holds 100 tons of molten iron, which is exactly the volume of a steel furnace that is fed from the crane. There are a certain number of steel furnaces, which work together and produce the final product of the steelworks. Figure 2 shows the schematic layout (not to scale) for the steelworks plant.

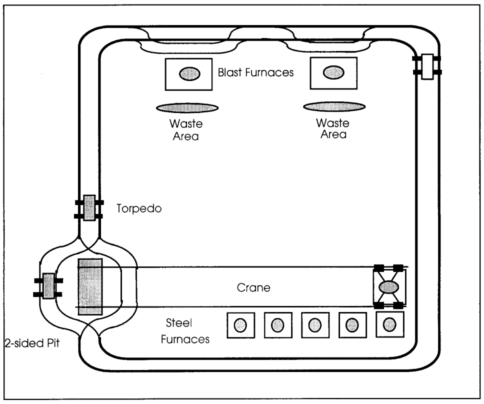

Figure 2: Layout for the Steelworks Plant

4.2 Input Parameters

As we can see from the brief explanation, the Steelworks model has several input variables. We are interested to know, in this particular simulation optimization problem, the optimum values of four of them: number of torpedoes, number of cranes, number of steel furnaces and the maximum volume that one torpedo could handle of molten iron. In our case all of the variables have integer values with resolution of 1. There is no restriction for the LEO Algorithm to handle real number, but there is no real gain if the optimum value of the volume of torpedo is 311.23 instead of 311. These input variables are depicted in table 1.

Table 1: Input Variables for the Steelworks problem

VARIABLE DESCRIPTION	MIN VALUE	MAX VALUE	RESOL. (STEP)
Number Of Torpedoes (Nt)	1	12	1
Number Of Cranes (Nc)	1	2	1
Number of Steel Furnaces (Nsf)	1	6	1
Torpedo Volume (Vt)	50	350	1

Hence a state is given by one combination of the input variables (Nt, Nc, Nsf, Vt). Note that the search space (all possible combinations of the variables) in this case is equal to 12*2*6*300 = 43200 points. This actually gives an idea of how difficult the optimization problem is if we try to obtain an optimum solution by evaluating all combinations of these variables.

4.3 Objective Function

The objective function in our case is basically the total cost of the Steelworks plant (Investment Cost and Operational Costs) calculated on a monthly basis. The operational cost is given by the volume of wasted melted iron. The investment costs could be calculated by the amortization of equipment (torpedoes, cranes and steel furnaces). Table 2 shows the amortization cost per unit of each equipment on a monthly basis.

Since the each ton of wasted melted iron costs approximately £100 it is possible to calculate the total monthly costs generated by the plant as shown in equation (1):

$$TMC = 2.1*Nt + 8.3*Nc + 16.7*Sf + 0.3*Waste \quad (£K) \qquad (1)$$

where "Waste" is the waste generated in 10 days of production. The TMC above, which was generated by the simulation model, was used as the input of the optimization procedure.

Table 2: Relative Equipment Prices

EQUIPMENTS	PRICE PER UNIT (MONTLY) (£)
Torpedoes	2,100.00
Cranes	8,300.00
Steel Furnaces	16,700.00

5 IMPLEMENTATION ISSUES AND RESULTS

The Steelwork models was built using Taylor II (Nordgren 1998) simulation software (v. 4.2) running on Windows NT operating systems on a Pentium II 300 MHz-based processor. The LEO algorithm was implemented in Visual Basic embedded within Excel version 97. The data exchange between Taylor II and the LEO algorithm was performed via an Excel Spreadsheet using DDE facilities. The front end of the optimization procedure is shown in Figure 3.

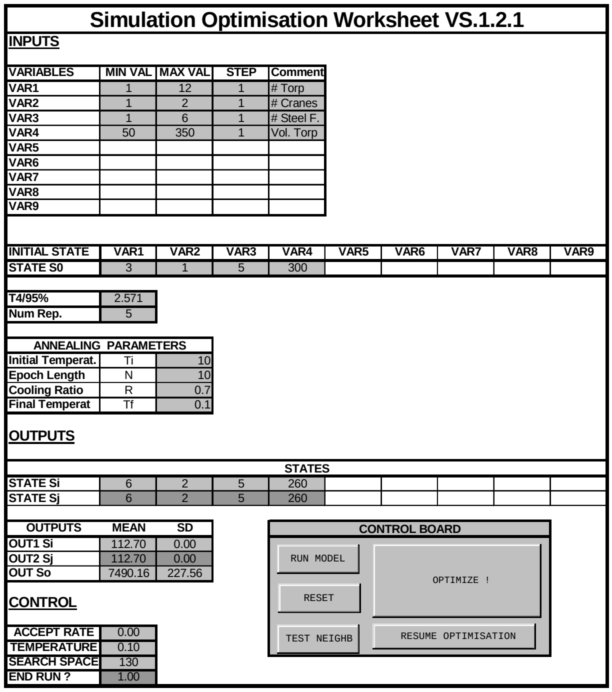

Figure 3: Front End of the Simulation Optimization Procedure (Excel Spreadsheet)

We simulated the operation of the plant during 12 days (considering the first 2 days as the "Warm-up period"). Five replications were generated for each combination of the variables (or states) and then we adopted a 95% confidence interval. The initial solution was 3 torpedoes, 1 crane, 5 Steel Furnaces and 300 tons of capacity for each torpedo. Results from the optimization are depicted on table 3.

Table 3: Simulation Optimization Results

VALUES	INITIAL SOLUTION	OPTIMUM SOLUTION *
Objective Function (K£)	7,490.16	112.70
Waste Generated (Tons/Month)	73,920	0
Number of Torpedoes	3	6
Number of Cranes	1	2
Number of Steel Furnaces	5	5
Torpedo Vol.	300	260

For this study the following annealing parameters were adopted: Initial temperature 10, final temperature 0.1, cooling Ratio = 0.7 and Epoch Length = 100. Note that the algorithm evaluated 130 possibilities which gives 130/43200 = 0.3% of the total search space. The solution adopted was optimum because we verified furthermore that the obtained number of resources is the minimum number that generates no waste. Another conclusion that we obtained (doing post optimization runs) is that the objective function (waste) is practically insensitive if the volume of a torpedo is higher than 250.

6 CONCLUSIONS

In this paper we have presented an application of the LEO optimization algorithm, in this case to a simulation optimization problem. LEO algorithm proved suitable for solving this type of problems (Hard Combinatorial and Stochastic). Paul and Chanev (1998) used a Genetic Algorithm to solve the same problem, but since the objective function they utilized differs from the one adopted here, we could not make a direct comparison. However they achieved their optimum values by exploring 0.4% of total Search Space which is very near to the one we achieved (0.3 %). This demonstrates that both algorithms (Simulated Annealing and GA) are equally powerful in solving problems of this nature.

In our future work we will try to dynamically vary the amounts of Move and Exchanges for LEO, to see if we can achieve improved performance and try to apply it to other kinds of optimization problems. Our intent is also for LEO to be freely distributed as a general tool for optimization.

ACKNOWLEDGEMENTS

This research was supported by CAPES (Brazilian Research Funding Agency), under reference no. 0439/98-8. Very Special thanks to the Department of Information Systems & Computing, Brunel University, who provided the resources necessary to this research, while Leonardo Chwif was a Research Visitor.

REFERENCES

Azadivar, F. (1992) Tutorial on Simulation Optimization, In *Proceedings of the 1992 Winter Simulation Conference*, Arlington, ed. J. Swain, D. Glodsman, R. C. Crain, and J. R. Wilson, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 198-204.

Barretto, M. R. P, Brito, N. M. J, Chwif, L., and Moscato, L. A. 1998 Scheduling with the LEO Algorithm, Technical Paper, Society of Manufacturing Engineers, 10-17.

Carson, Y. Maria, A. (1997) Simulation Optimisation: Methods And Applications, In *Proceedings of the 1997 Winter Simulation Conference*, Georgia, ed. S. Andradóttir, K. J. Healy, D. H. Withers, B. L. Nelson, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 118-126.

Chwif, L., Barretto, M. R. P. and Moscato, L. A. (1998) A Solution to the Facility Layout Problem using Simulated Annealing. *Computers in Industry,* 36:125-132.

Eglese, R. W. (1990) Simulated annealing: A Tool for OR, *European Journal of Operational Research*, 46(3): 271-281.

Fu, M. C. (1994) A Tutorial Review of Techniques for Simulation Optimisation, In *Proceedings of the 1994 Winter Simulation Conference*, Lake Buena Vista, ed. Jeffey D. Tew, S. Manivannan, D. A Sadowski, and A. F. Seila, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 149-156.

Glover, F. (1990) *Tabu Search for Call Routing – Part I*, Research Report, Graduate School of Business, University of Colorado, Boulder.

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Boston, Addison-Wesley Reading.

Harrell, C. and Tumay, K. (1994) *Simulation Made Easy: A Manager's Guide*, IE and Management Press, Norcross, Georgia.

Johnson, D. S., Aragon, C. R., McGeoch, L. A., and Schevon, C. (1989) Optimization by Simulated Annealing: An Experimental Evaluation; part I, Graph Partitioning, *Operations Research*, 37(6): 865-892.

Kirkpatrick, S., Gellat Jr., C. D., and Vecchi, M. P. (1983) Optimization by Simulated Annealing, *Science*, 220: 671-680.

Law, A. M. and Kelton, D. W. (1991) *Simulation Modeling and Analysis*, 2nd edition, McGraw-Hill Inc, New York.

Lee, Y. H. and Iawate, K. (1991) Part Ordering through Simulation-Optimization in an FMS. *International Journal of Production Research*, 29(7): 1309-1323.

Manz, E. M., Haddock, J., and Mittenthal, J. (1989) Optimization of an Automated Manufacturing System Simulation Model using Simulated Annealing, In *Proceedings of the 1989 Winter Simulation Conference*, ed. E. A. MacNair, K. J. Musselman, and P. Heidelberger, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 390-395.

Meketon, M. S. (1987) Optimization in Simulation: A Survey of Recent Results, In *Proceedings of the 1987 Winter Simulation Conference*, ed. A. Thesen, H. Grant, and W. D. Kelton, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 58-67.

Nordgren, W. B. (1998) Taylor II Manufacturing Simulation Software. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, Edward F. Watson, John S. Carson, and Mani S. Manivannan, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 263-267.

Paul, R. J. and Balmer, D. W. (1993) *Simulation Modeling*, Chartwell-Bratt, Lund.

Paul, R. J. and Chanev, T. S. (1998) Simulation Optimization Using a Genetic Algorithm, *Simulation Practice and Theory*, 6(6): 601-611.

Stuckman, B., Evans, G., and Mollaghasemi, M. (1991) Comparison of Global Search Methods for Design Optimization using Simulation, In *Proceedings of the 1991 Winter Simulation Conference*, ed. B. L. Nelson, W. D. Kelton, and G. M. Clark, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 937-943.

Tompkins, G. and Azadivar, F. (1995) Genetic Algorithms in Optimizing Simulated Systems, In *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopolous, K. Kang, W. R. Lilegdon, and D. Goldsman, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 757-762.

Van Laarhoven, P. J. M., and Aarts, E. H. L. (1987) *Simulated Annealing: Theory and Applications*, The Nerthelands, Reidel, Dordrecht.

AUTHOR BIOGRAPHIES

MARCOS RIBEIRO PEREIRA BARRETTO graduated in Electronics Engineering in 1983. He got his MSc degree in 1988 and his PhD in 1993, all from the University of Sao Paulo. He is currently Assistant Professor at the Mechanical Engineering Department of the Polytechnic School of the University of Sao Paulo. His research interests include manufacturing integration and CIM technology.

LEONARDO CHWIF graduated in Mechanical Engineering (Mechatronic Specialisation) in 1992 at the University of Sao Paulo and got his MSc degree in 1994 from the same University. He is currently a PhD student at the same University and he spent a period at Brunel University as a research visitor. Upon graduation Mr. Chwif joined the Brazilian branch of Mercedes-Benz truck manufacturer and then he joined the Brazilian branch of Whirlpool Corporation. His research interests includes manufacturing simulation and optimisation.

TILLAL ELDABI is a Researcher and Secretary to the Centre for Applied Simulation Modelling and a part-time Lecturer at the Department of Information Systems and Computing, all at Brunel University, UK. He received a B.Sc. in Econometrics and Social Statistics from the University of Khartoum and a M.Sc. in Simulation Modelling from Brunel University. His doctoral research is in aspects of healthcare management and the intervention of simulation. His main research concentrated on the economy of healthcare delivery. He is looking to exploit the means of simulation on the wider healthcare system management to assist in problem understanding.

RAY J. PAUL has been professor of Simulation Modelling at Brunel University since 1992, and Head of the Computer Science and Information Systems Department since 1993. He previously taught inforation systems and operational research for 21 years at the London School of Economics. He has published widely in book and paper form (two books, over 200 papers in journals, books and conference proceedings), mainly in the areas of the simulation modelling process and in software environments for simulation modelling. He has acted as a consultant for a variety of U.K. government departments, software companies, and commercial companies in the tobacco and oil industries. He is currently running a simulation research group, the Centre for Applied Simulation Modelling, made up of five faculty members and eight research students.