Belgian Journal of Operations Research, Statistics and Computer Science Vol. 35 (1) 95

The Integration of Optimization Paradigms and Simulation Practice in Industrial Management

H. Vanmaele
R. Van Landeghem

Dept. of Industrial Management
University of Ghent
Technologiepark 9
B-9052 Zwijnaarde (Belgium)

Abstract

Over the past decades, simulation has become one of the most widely used decision support techniques both in science and in industry. This increasing popularity is mainly caused by the ongoing performance improvements in hardware and software and by the growing maturity of simulation methodology.

Since the start of modern computer simulation practice at the end of the forties, simulation has mainly been used to model and to analyze the behavior of complex and non-deterministic systems, such as physical an biological systems, but also industrial processes, such as chemical reactors, manufacturing lines etc... The most important advantage of having a simulation model of such a system is that it allows for numerous experiments without interfering with the real system and its potential risks.

Although simulation as a methodology has no inherent optimization capabilities, the goal of simulation experiments is to enhance understanding of the system's behavior in order to optimize one or more system parameters (design) or variables (operation).

In this paper, an overview will be given of the options that are at the disposal of a simulation practitioner in the process of including optimization approaches in simulation projects.

INTRODUCTION

For many centuries, men have tried to capture features, appearance and characteristics of real-world complex systems into simplified models in order to run experiments and to predict the behavior of the real system under different conditions.

The history of simulation goes back for over 5000 years to the Chinese war games and this gives simulation common roots with several other Operations Research techniques which also found their origin in a military environment (Render, 1988). Since the 18th century, war games were used all over Europe to test out military strategies in simulated environments.

As for most other operations research techniques, the real conceptual importance of simulation as a problem solving technique was only appreciated as a side effect of the war efforts during World War II when Von Neumann developed Monte-Carlo simulation. This technique provided him with a quantitative framework to solve problems in physics that were too complex, dangerous or expensive to run in real environments.

The introduction of business computers in the 1950's and the development of specialized computer simulation languages (such as SIMSCRIPT and GPSS) in the 1960's gave an important impetus to simulation. It now became possible to handle quantitative models of large-scale problems in a more efficient way. The proliferation of computers and the ever-growing performance of hardware and software made simulation grow to become one of the most commonly used decision support and analysis tools in army strategy, scientific experiments, business and public management, education, ...

Although simulation in itself is a pure modeling tool and does not provide any real optimization paradigm, it is very often used in applications involving some kind of optimization. In most cases (e.g. in design), the simulation model is used to optimize a number of decision variables that determine the characteristics and the behavior of the system under study. Therefore, several approaches have been developed to accommodate for this need and to integrate simulation and optimization so that the resulting models combine the advantages of the two worlds.

In the first section of this paper, a short theoretical introduction will be given into the important concepts concerning simulation and optimization illustrating the need to combine both into one integrated methodology.

A general overview of different approaches to achieve this goal will be given in the second section. The most important methods will be discussed in some more detail throughout sections three (design of experiments), four (guided search), five (indirect optimization), six (dynamic optimization) and seven (hierarchical methods).

The last sections of this paper contain some general conclusions and references to related literature.

CHAPTER 1

Theoretical framework and concepts.

1.1. Simulation

Definition

Simulation is essentially the "art" of duplicating features, characteristics and appearance of a complex and/or non-deterministic system by building dynamic models that are used to run experiments under varying external conditions and operational or tactical decisions.

In most cases, simulation models will not be an exact replica of the original system. Features, characteristics and appearance will only be duplicated to the extent where they are of some *relevance* to the specific purpose of the model.

Simulation models exist in a large range of levels of detail and abstraction :

- pseudo-reality : e.g. army exercises which come very near to reality.
- small scale representation of the reality : e.g. pilot plants with a fraction of the nominal capacity of the real installation under study.
- analog models which consist of hydraulic, electrical or thermal equivalents of the real system.
- templates : iconographic abstraction of the real system using small elements (icons) which represent part of the system.
- computer models : the model is a computer program which duplicates the behavior of the real system.

Simulation is used only for systems that are either *complex or non-deterministic* (or both) and for which the use of a model is necessary because experiments on the real model would be either too dangerous or too expensive.

Complex systems are systems showing complex interactions between its elements in such a way that modeling its global behavior in one mathematical model (such as a mathematical programming model) seems to be impossible.

Non-deterministic systems behave according to random phenomena introduced by either external forces or by internal "noise" factors.

Simulation experiments will be run both to estimate the influence of *uncontrollable variables* and to understand the system behavior under different sets of *control policies* (fig. 1). The control policies can act both on the operational (implementation of control policies) or on the tactical level (use of resources). They are defined by the values of the decision variables which are exogeneous to the system (i.e. they are not influenced by the system itself).

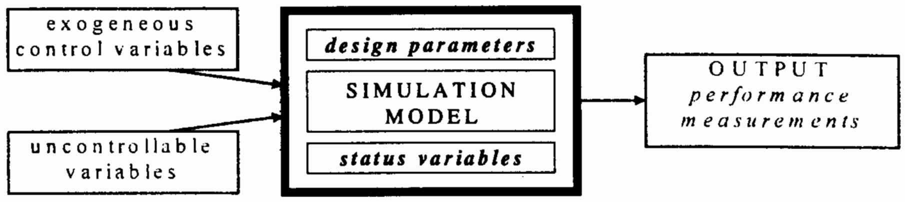

Figure 1 : Different types of variables in a simulation model

Monte Carlo simulation

The Monte Carlo method of simulation can be used for systems containing elements that exhibit randomness in their behavior. This technique can be applied in four steps :

1. Finding probability distributions for all stochastic variables included in the system. Techniques which can be used are : analysis of historical outcomes, sampling of the real system, expertise or intuition. The distributions can be empirical, normal, binomial, Poisson, exponential, ...
2. Transforming the probability distribution into a cumulative distribution.
3. Generating (pseudo-)random numbers (between 0 and 1) using random number tables or numerical algorithms such as linear congruential generators.
4. Running a simulation experiment which consists of :
   - generating random numbers.
   - matching the random number on the Y-axis of the cumulative distribution,
   - reading the corresponding value of the variable on the X-axis,
   - calculating the result using these values for each of the stochastic variables.

Discrete Event vs. Continuous simulation

Simulation science has evolved into two different directions. The first one handles continuous simulation. This means that the state of the model changes in a continuous way and that the model is built around a number of differential equations describing this continuous change of state. The equations are recalculated at fixed time increments through the simulation horizon. This type of simulation is mainly used to model physical and (bio-)chemical processes.
The second type of simulation is the discrete event simulation and is by far the most important one for business applications (manufacturing, administrative processes, logistics, ...). The events of interest (e.g. finishing a job, breakdown of a machine, arrival of an order ...) occur at discrete time points rather than continuously. The simulation time advances through the chronological list of events from one event to the next one, without any fixed time increments.

Computer simulation practice in business environments (Vanmaele, 1995)

A wide range of implementation options is available for building discrete event simulation models in business environments :

1. General purpose spreadsheets (Excel, Lotus, ...).
2. General purpose programming languages (Pascal, C/C++, ...).
3. Specialized simulation programming languages (Simula, Simscript, ...).
4. Graphical and interactive simulators (AutoMod, Witness, Arena, ...).

The main advantages to be expected from the use of simulation are :

- Risk reduction by analyzing the system behavior under different conditions.
- Experimentation without interference with the real system.
- Answering "what-if" questions and analyzing different alternatives on a common basis with equal random influences.
- Time compression.
- Introduction of very complex real world constraints.

However, the application of discrete event computer simulation in business environments has the following limitations :

- A "good" simulation model that has been sufficiently validated can require long development and testing times.
- Simulation does not guarantee to find optimal solutions to a problem. Therefore, we will discuss in the next sections the combination with optimization techniques.
- Building simulation models and designing experiments is a complex process that requires expertise and experience.

General structure of a simulation model

In general, a simulation model (fig.2) can be described as a black box calculation model transforming input data (parameters and variables) into output data (performance criteria).

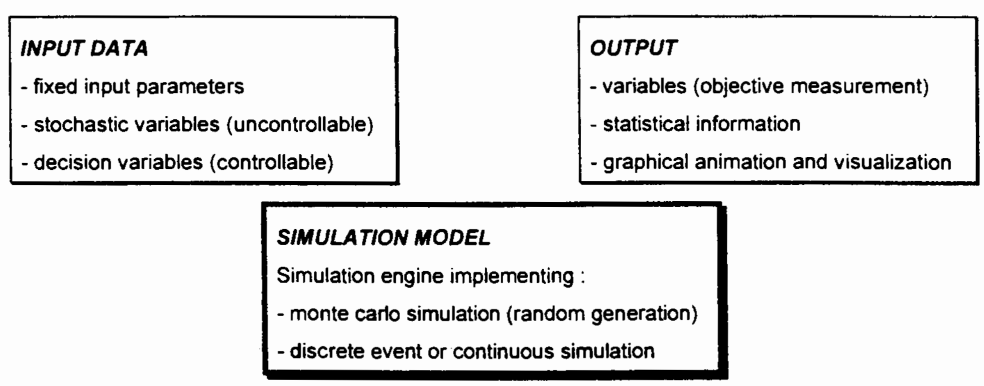

Figure 2 : General structure of a simulation model

1.2. Optimization

Definition

An optimization process consists of selecting from a set of alternative solutions the one solution that is better than all the other ones, using some type of evaluation or measurement criterion (objective function).

The wish and desire to improve the performance of an object, process or system is an essential expression of the human pursuit of perfection. A lot of industrial and research projects are the result of this drive and include some kind of optimization.

Mathematical formulation

The global optimization problem over a set S can be formulated as follows (in the case of a maximization problem) :

Given a set *S* that is a subset of R^n
and an objective function $f : D \rightarrow R$ and *S* is a subset of *D*,
find at least one point x^* that is an element of *S*
that satisfies $f(x^*) >= f(x)$ for every *x* being an element of *S*

- The minimization problem is included in this formulation because the minimization of $f(x)$ is equivalent to the maximization of $-f(x)$.
- Important theoretical work has been done to prove the existence and uniqueness of the global optimum for certain optimization problems. In practice, one will usually be interested in the complete set of solutions resulting in the same optimal objective value.

1.3. Simulation and optimization : focus of this tutorial

As mentioned in the previous paragraphs, a simulated system does not only include uncontrollable stochastic variables, internal state variables and fixed model parameters. In many cases, there are also some controllable decision variables involved which describe some kind of policy, investment option or operational rule that is to be applied during the process under study.

When simulating a system, one will want to use the results of the experiments in order to optimize the values of the controllable variables under random conditions determined by the uncontrollable variables. A secondary goal can be to enhance the robustness of the solutions. This means that we prefer solutions that are not sensitive to random influences introduced by the uncontrollable variables.

Several techniques exist to carry out the optimization task depending on the nature of the problem :

- Static optimization procedures are used to optimize controllable variables for which the optimal value is fixed over some period of time and does not depend on the status of the system at a certain point in time (e.g. investment decisions). Dynamic optimization, on the other hand, is used for decision variables that can be changed during the simulation depending on the model state at that time.
- The complexity of the problem determined by number of variables, type of variables, interaction effects, ... will determine which optimization methodology is most appropriate for a particular case.

In the following sections, an overview will be given of the major options that are available to solve the optimization of simulation models.

CHAPTER 2

General approaches to the optimization of simulation models.

2.1. Optimization of simulation models

The task of optimizing an objective function that is defined by the outcome of a model, such as a simulation model, can serve two different purposes :

- design of the model : tuning of model parameters.
- evaluation of experiments that are run on the model under different conditions.

In the first case, the model parameters are tuned and adapted in order to fit data measured on the real system. The objective is to minimize the deviation between simulated and measured data. This can be a very useful approach during the validation phase in a simulation project.

In the case of evaluation of alternatives on a simulation model, the objective is to find values for the controllable decision variables that maximize (minimize) a quantitative criterion. This quantitative criterion is calculated for every alternative taken into consideration by running a simulation experiment.

In the following paragraphs, we will focus on the case of evaluation of alternative experiments run on a simulation model. However, the methods and tools that are described can be used just as well in the case of model design.

The only difference is in the input data that are fixed and the ones that can be changed in order to optimize some objective function :

	variable input data	**fixed input data**	**objective function**
model design	model parameters and probability distributions describing the system	all other input data such as controllable decision variables	deviation between simulated and measured data
evaluation of alternatives	controllable decision variables describing an operational or tactical policy	model parameters and probability distributions describing the physical system	quantitative measurement criteria describing system performance

2.2. General optimization approaches

Enumerative Search : Design Of Experiments (D.O.E.)

The D.O.E. methodology is an optimization approach that is based on running experiments for a set of different alternatives. This set of alternative solutions is defined before the actual experimentation on the simulation model is started. This means that the result from a certain experiment is not used in selecting the next experiment. A very well known scheme to select the set of alternatives is the full factorial design scheme. Other (partial) schemes such as the Taguchi D.O.E. techniques will be discussed in section 3.

Guided Search

This approach is also based on the evaluation of alternative solutions. but the complete set of alternatives to be taken into consideration is not defined in advance. The selection of alternatives for experimentation is based on the results of previous experiments. The rules that control this selection process can be divided in several classes :

- numerical methods that do not require derivatives,
- gradient based methods,
- random search methods.

Indirect optimization

The basic approach behind all these methods is to use (local) information to build a meta-model for the simulation model. This meta-model includes an approximation of the input-output relations of the simulation model and is built in such a way that it is easier to optimize than the underlying simulation model.

Dynamic integrated optimization based on local decision rules

This approach is useful when the variables to be optimized are not static variables that have the same value throughout the simulation experiment (e.g. design variables such as layout parameters) but dynamic variables (e.g. operational variables such as the number of resources to be used at a certain time). In the latter case, we are not looking for one value for a particular variable but for the value of a certain variable at every point in time.

Hierarchical optimization approaches

The hierarchical approach is based on the combination of simulation and an optimization method and is not a real integration of both paradigms. The optimization methods used in this approach are in most cases based on a programming method such as constraint logic programming (CLP) or mixed integer linear programming (MILP).

2.3. Basic techniques to reduce the complexity of the optimization problem

Simulation models very often have large numbers of parameters, describing and defining the system, and variables controlling the system. Furthermore, some of these parameters and variables (factors) will have very large ranges of acceptable values. The combination of these two facts results in very difficult combinatorial optimization problems. Therefore, it will often be necessary to reduce the complexity of the problem at hand :

1. ***Reduction of the number of levels taken into consideration for each factor.***
   For continuous factors, a few relevant discrete levels are to be chosen in such a way that all critical values (where important changes occur) are included. The number of levels considered for factors that are already discrete, will have to be reduced to include only the relevant and critical levels.
2. ***Reduction of the number of factors taken into consideration in the optimization problem.***
   In most cases, the behavior of the system under study will only be influenced by some key parameters and variables while the others only have a marginal effect on the system. Therefore, it will often be preferable to include only these key influences into the optimization problem.

Although this complexity reduction can be done based on intuition, experience, previous research or very simple gradient methods, we will discuss here some more advanced techniques that can be useful in analyzing the importance of factors and their level values.

Frequency domain methodology (FDM) (Jacobson et al., 1991)

During a simulation experiment, the values of the factors that are selected for analysis are changed according to a sinusoidal oscillation with a unique frequency for each factor. If the system's output is particularly sensitive to a certain factor (this means that we have an important factor), then this oscillatory change will induce similar oscillations in the output.

By oscillating a number of factors each with a different frequency during a simulation experiment and by analyzing the oscillations found back in the output, it is possible to determine the importance of each factor.

Group Screening techniques (Kleijnen, 1992)

This basic method consists of partitioning the different factors in a limited number of groups that contain related factors of the same type. A number of experiments is then set up where each group is considered as one factor. This means that all factors in a group change in the same way from one experiment to another. It can be proven that if the main effect of a certain group is not important, no factor belonging to this group is important on its own. Several variants of this method exist for particular situations.

CHAPTER 3

Design Of Experiment Techniques (DOE).

3.1. Introduction

Traditional methods for the optimization of static variables in simulation models are mostly based on the evaluation of a number of alternatives that are selected from the complete set of possible solutions (i.e. the search space).

DOE-techniques provide a way to set up the complete experimental design (i.e. selection of the alternatives that will be evaluated) before the experimentation process begins. The evaluation points are chosen in order to cover the search space as completely as possible. This approach has two major advantages :

- The number of experimental runs is known in advance and does not depend on the outcome of the experiments.
- If the number of evaluation points is large enough, there is no risk of getting stuck in a local optimum.

The main disadvantages are :

- Information of previous experiments is not used in setting up the next experiment, so a lot of time can be lost in evaluating areas of the search space that are not at all interesting.
- The required number of experiments can be very large in order to get acceptable results.

DOE-methods can in general only be applied to discrete variables, so the first step before applying a DOE-method consists of choosing a limited number of discrete values in the domain of each continuous variable. In choosing these discrete values, one has to make sure that all relevant values are included.

Several schemes for setting up experimental designs are known from literature. We will discuss three in some more detail : "one factor at a time" - design, full factorial design and Taguchi design.

3.2. One factor at a time design

For this type of design, the first experiment is run for a base configuration of factor levels. Additional experiments are run for configurations where, compared to the base configuration, only one of the factors is changed. Each such additional experiment allows for estimating the influence of the particular factor that was changed during the experiment.

This approach starts from the assumption that the different factors are independent, i.e. no interactions exist.

3.3. Full factorial experimental design (exhaustive search)

The method itself consists of running a simulation experiment for every possible combination of values for each of the decision variables. For a system including 4 variables each having 3 possible values, this requires 81 experiments. By this technique, the number of alternatives to evaluate can become very large. On the other hand, this method guarantees to find the global optimum in the case of discrete variables and an approximation of the global optimum in the case of continuous variables on condition that the number of discrete points in the continuous domains are chosen sufficiently large.

Furthermore, this method takes into account any type of interaction that could be present between two ore more factors in the system.

3.4. Taguchi design (Roy,1990)

Several techniques exist to set up partial factorial experiments including only a few of the possible combinations of decision variable values. One interesting technique is the Taguchi design of experiments. The first step is to rank the n relevant values of each decision variable and give them a level-number from 1 to n. The next stage is to setup the experiments. This is done using specially constructed orthogonal arrays containing a number of rows. Each row defines one experiment to be carried out with the corresponding levels for the variables.

This procedure can dramatically reduce the number of experiments that is needed, e.g. for a problem with 7 decision variables with each of them two relevant values, a full factorial experiment would require 128 simulation experiments, the Taguchi design reduces this number to 8. Special facilities are offered to study the interaction between variables and the significance of the results.
An example of this approach for a simulation based scheduling model can be found in (Vanmaele and Van Landeghem, 1994)

CHAPTER 4
Guided search methods

4.1. Introduction

"Design of experiment" - techniques are used to set up the factor values in each experiment before the real simulation experiments start. The search methods discussed in this chapter are also based on the principle of changing certain factor values and analyzing the effect of these changes on the system's output. However, in guided search methods, the result of the previous experiment(s) is used to decide on the factor values that will be changed to run the following experiment.
The general idea behind this principle is that by using information from previous runs, we will be able to set up the experiments in a more intelligent way so that parts of the search space which are not interesting in terms of optimum seeking are not used for running experiments.
In this section we discuss three classes of guided search methods, depending on the search algorithm used : numerical methods not requiring derivatives, gradient based methods and random search methods based on metaheuristics.

4.2. Numerical methods not requiring derivatives

In this category, we find very well known methods such as the Gauss-Seidel and the Hooke-Jeeves method (Kuhn and Wiwat, 1994). These methods are based on the evaluation of the neighborhood of a certain starting point (by using finite differences) in all coordinate directions in order to find a "best" direction to move to the next evaluation point. These methods can only be used for the optimization of rather simple problems.

4.3. Gradient based methods

These methods are in general based on the calculation of gradients (by an analytical approach or by numerical approximation) in order to move through the search space while increasing the value of the goal function. In general, these methods are not very useful for the optimization of simulation models since input-output relations of simulation models can be very complex and do not behave in such a way that derivatives have any relevant meaning or can easily be calculated.

4.4. Random Search methods based on metaheuristics (Szczerbicka, 1994 / Pirlot, 1992)

Guided random search methods have been developed in order to handle the ***global*** optimization of simulation models with the following characteristics :

- the goal function is very complex (nonlinear, multimodal, multidimensional, influenced by random effects, noncontinuous, nondifferentiable, ...),
- the search space and the number of possible solutions are very large.

Classical optimization methods are not capable of handling this kind of optimization problem because of a number of reasons :

- most of them (except for the enumerative methods) are local in scope : they look for an optimum in a neighborhood of a starting solution.
- a lot of the classical calculus-based methods assume the existence of derivatives or some numerical approximation of derivatives.
- the enumerative methods are too time consuming.

Guided random search methods such as simulated annealing and genetic algorithms (evolution strategies) are directed search methods based on stochastic processes that are found back in nature. The stochastic nature of the methods is essential for not getting stuck in local optima and for eventually finding global optima.

Simulated Annealing (Van Laarhoven and Aarts, 1987)

The simulated annealing metaheuristic is based on the physical process of cooling solids to low energy states. To obtain these low energy states (minimum of the energy function) it is necessary to heat the material to very high temperatures at which the material can undergo random changes in its structure. By gradually cooling the material, the structure will be frozen in the way that corresponds to the lowest energy state.

This principle is transferred to optimization problems in the following way (for a minimization problem) : starting from an initial point in the solution space, a new evaluation

point is randomly chosen in its neighborhood. The goal function is calculated for this new point. If the value is lower than the value for the previous point, then the new one is accepted and used as the new starting point ; if the new value is higher, then it is accepted with a certain probability that decreases as the optimization (cooling) process proceeds. Accepting worse solutions with a certain probability makes it possible to escape from local optima.
Experience with this method has shown that it is very easy to implement and can give very satisfactory results. The required optimization time, however, can be (too) long.

Genetic algorithms (Goldberg, 1989)

The underlying concept of genetic algorithms or evolutionary algorithms more in general, is that of natural evolution (natural selection, "survival of the fittest").

In order to apply a genetic algorithm, the elements of the solution space have to be represented as strings of binary entities. Starting from an initial population (parent population) of solutions (set of binary strings), a new population (second generation) is generated by some basic genetic mechanisms such as mutation, inversion and crossover. All solutions belonging to this new generation are evaluated according to some objective function in order to find the "fittest" among them. These are used to generate a new (third) generation and this process is repeated as long as necessary (according to some terminating condition).
In general terms, using these algorithms, we count on the fact that the average "fitness" of each generation will be larger than that of its parent generation and that there will be some type of convergence in order to find global optima for the problem.

Implementation of this methods is more difficult because of two reasons :

- Representation of the solutions as binary strings is essential for the application of the method.
- The techniques for generating new generations can be very different from one implementation to another and can require a lot of fine-tuning to obtain fast convergence.

However, our experience in using this method for the optimization of simulation models is very promising and genetic algorithms should certainly be considered as one of the optimization methodologies for the future.

CHAPTER 5

Indirect optimization : Response Surface Method (RSM), Frequency Domain Methodology (FDM).

5.1. Response Surface Method (RSM)

Response Surface Methods (Myers et al., 1989) are based on building a global (meta-)model for the goal function implemented in a simulation model. Building this response surface is done using local information. Optimization is achieved by setting the gradient of the goal function represented by the response surface, equal to zero.

These methods can only be used successfully if a number of assumptions are fulfilled, which is very often not the case in practical situations. This makes these methods not generally applicable. Furthermore, the method can require very difficult and complex mathematical operations.

One of the basic techniques in RSM consists of building a polynomial (first and second order) model to describe the goal function. In most cases, a first order model will be used until a (local) optimum is reached. At this point, a second order model is chosen to find the exact optimum. Optimization of the polynomial models is gradient based. This approach is of course a local optimization and its effectiveness is mainly depending on the nature of the real goal function that is to be optimized.

In some cases, the Taguchi method (cfr. section 3) for optimizing simulation models is also considered to belong to the class of indirect optimization based on metamodels because the construction of the Taguchi orthogonal arrays is based on some assumptions about the simulation model. This means in fact that some kind of metamodel is assumed for the simulation model under study.

5.2. Frequency Domain Methodology (FDM)

The Frequency Domain analysis (Jacobson et al., 1991) consists of running one (or more) experiments in which some of the decision variables or system parameters are oscillated in a sinusoidal way. However, a unique oscillation frequency is assigned to each of the variables or parameters under study. The method is based on the following assumption : if the system's response is sensitive to a particular variable or parameter, the oscillation of this variable or parameter will induce similar oscillations with the same frequency in the response. Analysis of the oscillation frequencies found in the response function will give important information on the sensitivity of the system to a particular parameter or variable and on the optimal value of these values.

CHAPTER 6

Dynamic Optimization based on Local Decision Rules

6.1. Introduction

In the case of dynamic optimization, the objective is to optimize one or more dynamic variables during the simulation time. The value of a dynamic variable can change during the simulation when some type of event affecting this particular variable occurs. At such a decision point, an optimal value for the decision variable is selected from a set of possible values based on a heuristical procedure or some other optimization technique. This is of course a local and very myopic type of optimization (even when using decision rules with lookahead capacity) resulting in a sub-optimal solution for the overall system.
A typical example of this approach is found in simulation based scheduling.

6.2. Simulation based scheduling (Vanmaele, 1995)

Scheduling is the process of allocating resources such as machines, personnel, tools and components over time to perform a collection of tasks. A good schedule takes into account all constraints that exist within the real environment and is a compromise between different performance goals such as meeting due dates, decreasing work in process, etc...
One way of producing schedules is by using a discrete event simulation model.
This approach requires that one type of resources is selected as the primary resource type, the other resources are defined as secondary resources. The system is then simulated using the discrete event model until one of the primary resources has finished its job and becomes idle. This is the event causing a decision point : at this time, the resource has to decide on the next job from its waiting queue that is to be processed. This decision is made using task selection or dispatching rules which evaluate the jobs in the waiting queue and order them according to some set of criteria. Task selection or dispatching rules can be thought of as a series of filters through which potential candidate jobs are fed. The filtering process narrows the set of candidates until one or no jobs remain to be selected. Each filter layer is a criterion or test that must be satisfied by the selected job. These criteria can be very local (due date, user priority, processing time) or more advanced and dynamic such as the number of jobs down-stream in the same routing or some global status variable of the system.
The advantage of simulation based scheduling is that it can produce schedules at finite capacity for systems that can have very complex constraints or logical interrelations since all constraints and logic are part of the simulation model which can be as complex as necessary.
The limitations of simulation based scheduling are in the fact that this technique is essentially based on forward scheduling (no backward or bottleneck scheduling) and on local optimization of the schedule (optimization is done each time an event occurs and is in most cases very myopic).

CHAPTER 7

Hierarchical approach : Simulation and Constraint Logic Programming

7.1. Constraint Logic Programming (CLP)

Constraint programming (Fron, 1994) is a new technology that combines elements from operations research and artificial intelligence. Solving a problem using CLP requires a number of steps :

- definition of variables and their domains,
- declaration of relations between the variables, i.e. the constraints of the problem,
- propagation of these constraints in order to reduce the domain of the variables,
- generation of solutions that satisfy the constraints by the following process :
  - instantiation of one variable (reduction of the domain to one value),
  - propagation to reduce the domain of other variables,
  - if consistent (no domain becomes empty) : instantiate another variable,
    else : backtracking,
  - until all variables are instantiated.

This process is designed to find acceptable solutions satisfying all constraints and not to find optimal solutions optimizing some type of objective function. Therefore, it will often be necessary to add an optimization step based on an incremental process such as branch and bound.

7.2. Hierarchical combination of simulation and constraint programming

Using a hierarchical approach, it is possible to combine the strengths of both discrete event simulation and constraint programming.

Simulation is especially useful to model complex, non-deterministic systems. However, the dynamic optimization process described in section 6 is local and does not include any type of backtracking : once a decision is taken based on local information, it cannot be undone. This results in sub-optimization.

Constraint programming can handle complex combinatorial problems and uses global information and backtracking in the optimization process. The same advantages but under some different limitations are valid for general mixed integer linear programming.

The combination of these two approaches (simulation and optimization) is done in a hierarchical way. This combination, which is not a real integration, can be achieved in two different ways :

- A simulation model with dynamic integrated optimization can be used to set out the rough guidelines for the optimal solution. Another optimization technique such as (mixed integer) linear programming or constraint logic programming is then used to optimize this solution within the guidelines that were defined by simulation (sub-optimization).

- The simulation model is used to get more details about a solution that has been optimized at a higher level using heuristics, expert knowledge, programming methods such as MILP or CLP, etc... The higher level optimization will for example decide on the sequence of a number of jobs based on aggregate resource utilization, the simulation is then used to calculate the exact number of resources used and to put time stamps on the different jobs.

CONCLUSIONS

During the past decades, simulation has become one of the most widely used decision support techniques in business environments. Discrete event simulation models are in the first place used for understanding, analyzing and visualizing complex and non-deterministic processes that can be found in almost every business problem of some importance.
This analysis function, however, is very closely related to optimization issues such as reduction of queues, generation of schedules, investment decisions, definition of facility layouts etc... Therefore, it has become more and more necessary to look for options to integrate optimization into the simulation methodology.

In this paper, an overview is given of the available approaches and some of them are discussed without really going into details. The most important ones are :

- Design of Experiments (DOE) such as full factorial and Taguchi design,
- Guided Search Algorithms based on gradient methods, numeric methods (e.g. Gauss-Seidel) and metaheuristics (Genetic Algorithms, Simulated Annealing),
- Indirect Optimization such as the Response Surface Method (RSM),
- Dynamic Integrated Optimization as is used in simulation based scheduling,
- Hierarchical approaches used in combination with e.g. constraint logic programming.

Furthermore, some general techniques for analysis of the behavior of the simulation model and for the reduction of the complexity of the optimization problem were discussed. The two main techniques are :

- Frequency Domain Methodology (FDM),
- Group Screening Techniques.

The further development of optimization techniques for simulation models together with further increases in performance, object orientation, GUI's and functionality of simulation software will undoubtedly contribute a great deal to a continuing increase in popularity of simulation methodology as a decision support technique in business environments.

REFERENCES

Banks, J. and Carson, J.S. ; ***Discrete Event Systems Simulation*** ; Prentice-Hall ; 1984.

Fron, A. ; ***Programmation par contraintes*** ; Addison-Wesley ; 1994 ; 393 p.

Goldberg, D.E. ; ***"Genetic algorithms in search, optimization and machine learning"*** ; Addison Westley, Reading, MA ; 1989.

Jacobson, S. , Buss, A. and Schruben, L. ; ***"Driving Frequency Selection for Frequency Domain Simulation Experiments"*** ; Operations Research 39.6 ; pp. 917-924.

Kleijnen, J. and Van Groenendaal, W. ; ***Simulation, a statistical perspective*** ; J. Wiley and Sons ; 1992 ; 241 p.

Kuehn, W. and Wiwat, P. ; ***"Parameter optimization in discrete event simulation"*** ; proceedings of Worksims 94 : simulation in manufacturing systems, edited by Kuehn and Nagarur, Bangkok, 1994 ; pp. 151-154.

Law, A.M. and Kelton, W.D. ; ***Simulation Modeling and Analysis*** ; McGraw-Hill ; 1991.

Myers, R.H., Khuri, A.I. and Carter, W.H. ; ***"Response Surface Methodology : 1966-1988"*** ; Technometrics, vol. 31 ; 1989 ; pp. 137-157.

Pirlot, M. ; ***"General local search heuristics in combinatorial optimization : a tutorial"*** ; JORBEL, vol. 32, no. 1-2 ; 1992 ; pp. 7-68.

Render, B. and Stair, R.M. ; ***Quantitative analysis for Management*** ; pp. 578-625 ; Allyn and Bacon ; 1988.

Roy, R.K. ; ***A primer on the Taguchi method*** ; Van Nostrand Reinhold ; 1990.

Szczerbicka, H. ; ***"Are Genetic Algorithms a Panacea for Optimization of Simulation Models ? "*** ; proceedings of the conference on modeling and simulation ESM'94, edited by Guasch and Huber, Barcelona, 1994 ; pp. 38 - 45.

Van Laarhoven, P.J.P. and Aarts, E.H.L. ; ***Simulated annealing : theory and applications*** ; D. Reidel Publ. Co ; Dordrecht : 1987.

Vanmaele, H. ; ***"Simulation in Industrial Management Applications"*** ; CIM_CIL Technology Transfer Center Ghent, seminar text, 1995.

Vanmaele, H. , Van Landeghem, R. ; ***"Using the Taguchi method in the design process of scheduling policies"*** ; proceedings of the conference on modeling and simulation ESM'94, edited by Guasch and Huber, Barcelona, 1994 ; pp. 888 - 891.