Iterative simulation optimisation approach-based
genetic algorithm for lot-sizing problem in
make-to-order sector

Ahmed Ammeri*
Unit of Logistic, Industrial and Quality Management (LOGIQ),
High Institute of Industrial Management of Sfax,
B.P. 954, 3018, Sfax, Tunisia
E-mail: ammamiahmed@yahoo.fr
*Corresponding author

Wafik Hachicha
Unit of Mechanic, Modelling and Production (U2MP),
Engineering School of Sfax,
B.P. 1173, Sfax 3038, Tunisia
and
Department of Industrial Management,
High Institute of Industrial Management of Sfax (ISGI),
B.P. 954, Sfax 3018, Tunisia
E-mail: wafik_hachicha@yahoo.fr

Habib Chabchoub
Unit of Logistic, Industrial and Quality Management (LOGIQ),
University of Economic Sciences and Management,
B.P. 954, 3018, Sfax, Tunisia
E-mail: habib.chabchoub@fsegs.rnu.tn

Faouzi Masmoudi
Unit of Mechanic, Modelling and Production (U2MP),
Department of Mechanical Engineering, Engineering School of Sfax,
B. P. W, Sfax 3038, Tunisia
Fax: +216-74-274-088
E-mail: faouzi.masmoudi@enis.rnu.tn

Abstract: This paper describes the development of a process simulation model and integration of the genetic algorithms (GA) with the model as optimisation techniques using a case study of lot sizing problem (LSP) in make-to-order (MTO) supply chain solved by a combined simulation and genetic algorithm (GA) optimisation model. The simulation model is performed using ARENA software. GA model is implemented using visual basic for application (VBA) language because it ensures exchanges between ARENA software and MS Excel. The case study’s objective is to determine the optimal solution to determine the fixed lot size for each manufacturing product type that will
ensure order mean flow time target. The comparative results with OptQuest software, which is used as a global search method, illustrate the efficiency and effectiveness of the proposed approach.

**Keywords:** lot sizing problem; LSP; make to order supply chain; simulation optimisation; genetic algorithm; VBA; case study.


**Biographical notes:** Ahmed Ammeri is a PhD student at University of Economic Sciences and Management of Sfax in Tunisia (FSEG). He obtained his Master of Science in Transport and Logistic in 2009. He is a member of LOGIQ Unit. His research activities deal with the optimisation and supply chain management.

Wafik Hachicha is an Industrial Engineer (1999) from ENIT Tunisia. He obtained his PhD in Manufacturing Management in 2009. He is a researcher at the Mechanics Modeling and Production Research Unit (U2MP). His research activities deal with the modelling, analysis, optimisation, and the simulation of manufacturing and supply chain systems. He is an Assistant Professor at the High Institute of Industrial Management of Sfax in Tunisia (ISGI).

Habib Chabchoub is a Professor in Quantitative Methods at the University of Economic Sciences and Management of Sfax in Tunisia, and the Director of the Laboratory GIAD (Gestion Industrielle et Aide à la Décision), Sfax, Tunisia. He is a member of Logistic, Industrial and Quality Management (LOGIQ) Unit.

Faouzi Masmoudi received his PhD in Computer-Integrated Manufacturing (1988) from ENSAM de Paris, France. He is a Professor at the National School of Engineers of Sfax in Tunisia and a Researcher at the Mechanics Modeling and Production Research Unit (U2MP). His research activities are modelling and simulation of manufacturing systems; design and layout of the cellular systems of production and simulation of manufacturing cells with unreliable machines.


## 1 Introduction

Supply chain (SC) is a network of suppliers, manufacturers, distributors, and retailers that act together to control, manage, and improve the overall SC performance (Abd El-Aal et al., 2011). SC optimisation involves making decisions for proper organisation, planning and design of production, storage locations and transportation of SC which are vital for retaining the competitive edge of companies in a global economy. These problems are often very large and complex due to the large number of facilities of the SC
such as the number of plants, warehouses and retailers, and due to complex interactions among these facilities such as the modes of transport, relocation of warehouses, the stochastic nature of the demand, etc. The optimisation of SC often deals with the situation in which the interest is to find which of the large number of sets of model specifications lead to optimal output performance (April et al., 2003). SC has been traditionally modelled in analytic ways using deterministic or stochastic methods in order to implement supply chain management (SCM) in real logistic world. Simulation is preferred when an analytic solution cannot give proper values for performance evaluations. Simulation generates numerical and logical models based on real-world problems and imitates different scenarios using computers to find solutions for problems. There are some studies to advance the activities of simulation. Bottani and Montanari (2011) presented a simulation tool for the design and performance analysis of supply networks. The tool, developed under MS Excel and properly set up for this purpose, reproduces a fast moving consumer goods (FMCG) supply network, and incorporates viable input data taken from previous studies in that field and they analysed four network configurations, stemming from the combination of different numbers of echelons and of facilities per echelon. Furthermore, Bruzzone and Bocca (2012) proposed a new approach based on modelling and simulation to support decision making in marine logistics. The aim is to identify innovative and effective solutions for maritime logistics problems with a particular focus on petrochemical industries, where different production sites need exchange intermediate products by marine lines the authors propose a simulation model, named maritime logistic network (MARLON) which is combined together with additional simulators, developed by the authors, in order to support decision makers in maritime logistics. Moreover, Detlef Herbert and Valverde (2012) examined the performance of the SC by using a simulation with a stock out penalty and the percentage of items delivered from stock. Results from simulation show that no single solution for the reorder point and quantity with an optimal stock out penalty can be found. The simulation generated a solution for the minimum average penalty.

Simulation models require fewer restrictive assumptions than mathematical models when representing complex, dynamic systems. These models themselves are usually fairly complex and of relatively high dimensionality. That is, the performance of a simulation model mostly depends on a large number of parameters or factors that act and interact in a complex manner. However, with simulation modelling, the relationships between the design parameters and their resulting performance measures are not explicitly known. For complex problems with high risks or that are impossible for real-world testing, computer simulation technology provides an effective tool to help plan for solving, analysing and evaluating different alternatives. Computer hardware and software are continually improved and updated, and computer simulation software is high in computation efficiency and accuracy. However, computer simulations sometimes had not been considered as an ideal tool for problem solving because conventional computer simulations used the exhaustive method, in which all possible solutions need to be find into the simulation system to compute the best parameters setting and resources combination. This method is not only very time-consuming and ineffective, but also costly. As a result, recently many researchers have purposed that by integrating optimisation algorithms, computer simulation can determine the best combination more quickly and efficiently. In addition to a good model, one also needs a sound technique to utilise the information from a simulation to make a decision. One such technique is optimisation via simulation. SO requires the evaluation of a simulation model in the form
Iterative simulation optimisation approach-based GA for LSP in MTO sector

of responses to a ‘What if’ question. ‘What if’ questions demand answers on certain performance measures for a given set of values for the decision variables of the system. Recently, the advancement of computer technologies enables us to answer to ‘How to’ questions as well. ‘How to’ questions seek optimum values for the decision variables system so that a given response or a vector of responses are maximised or minimised. So is an active area that has deserved a special attention from both simulation researchers and practitioners.

Thus, the idea in this paper is to illustrate the hybrid approach for solving the lot sizing problem (LSP) in make-to-order (MTO) SC. We propose a hybrid simulation optimisation approach that combines both the optimisation and the simulation modelling approaches. This study integrates simulation techniques with a genetic algorithms (GA) using an iterative process. GA model is implemented using visual basic for application (VBA) language. To demonstrate the feasibility of this approach, the hybrid approach is applied to LSP in MTO supply chain. The optimal solution is to determine the fixed lot size for each manufacturing product type that will ensure order mean flow time (OMFT) target and the results are discussed.

The rest of the paper is organised as follows: Section 2 presents a brief review of the literature on simulation optimisation methods. In Section 3 we present a complete case study description, which is used as an illustration. Section 4 presents the developed optimisation and simulation approach based on GA and the proposed hybrid method is explained in detail. Section 5 summarises the results of the optimisation performed and Section 6 is seen as a conclusion.

2 Related literature reviews

2.1 Genetic algorithm

GA were invented by John Holland in 1960s and further developments were made by the mentioned scientist and a number of his students and colleagues at the University of Michigan between the 1960s and the 1970s.

GA are search algorithms based on the mechanics of natural selection and natural genetics. They combine survival of the fittest amongst string structures with a structured yet randomised information exchanges to form a search algorithm with some of the innovative flair associated with human search (Goldberg, 1989). GA are powerful and broadly applicable stochastic search and optimisation techniques and are perhaps the most widely known types of evolutionary computation methods.

The basic algorithms in GA consist of selection, crossover and mutation operators, which are the so-called genetic operators. GA searches the solution space by building and evolving a population of solutions. The evolution is achieved by means of mechanisms that create new trial solutions from the combination of two or more solutions that are in the current population. Transformation of a single solution into a new trial solution is also considered in these algorithms. The main advantage of evolutionary approaches in general is that they are capable of exploring a larger area of the solution space with a smaller number of objective function evaluations, provided that they are implemented effectively.
The GA maintains a population of individuals for each generation. Each individual represents a potential solution to the problem under investigation. Each individual is evaluated to give some measure of its fitness. Some individuals undergo stochastic transformations by means of genetic operations to form new individuals.

There are two types of transformation which are mutation and crossover. The first creates new individuals by making changes in a single individual, and the second creates new individuals by combining parts from two individuals. New individuals, called offspring, are then evaluated. The operation of the GA is shown in Figure 1.

**Figure 1** Genetic algorithm procedure

- Population initialisation
- Population evaluation
- Selection
- Crossover
- Mutation
- Population evaluation and updating
- Stopping criterion
  - Yes: Best solution
  - No

A new population is formed by selecting the most fit individuals from the parent population and the offspring population. After several generations, the algorithm converges to the best individual. This hopefully represents an optimal or suboptimal solution to the problem (Gen and Cheng, 2000).

GA are implemented in a computer simulation in which a population of abstract representations called chromosomes or the genotype of the genome of candidate solutions called individuals, creatures, or phenotypes to an optimisation problem gives better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness) and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.
2.2 Simulation-based GA

The computer simulation is just a model or a function that transforms the inputs into the outputs. The operational parameters and their variables are described as the inputs and the performances, which are derived from simulation, are described as the outputs. The operational conditions are then tested on this model to achieve the objectives. One objective of the application of simulation is to search for a set of operational parameters so that system performance is improved.

Simulation is essentially a trial-and-error approach. It is merely a tool for problem solving that cannot provide an answer. In addition to a good model, one also needs a sound technique to use the information from a simulation to make a decision, one such technique is SO.

SO provides a structured approach to determine optimal input parameter values, where optimal is measured by a function of output variables associated with a simulation model. Several excellent surveys have been written on this topic. Methods for SO vary greatly depending on the exact problem setting. Different classifications used by various researchers can be found in Carson and Maria (1997), Tekin and Sabuncuoglu (2004) and Andradóttir (2006). These surveys classified the existing techniques according to problem characteristics such as Random search and metaheuristics approaches, ranking and selection, direct and indirect gradient estimation and metamodel methods. Furthermore, Ammeri et al. (2010) presented a general overview of the different approaches for simulation optimisation with all classification criteria found in the research literature and include pointers/references to the state of the art. They have classified techniques for SO into four groups:

1. statistical selection methods
2. metamodel methods
3. stochastic gradient estimation
4. global search methods.

An overview of the extended classification scheme is shown in Figure 2. Therefore, a literature affectation has been presented since 1995 up to 2010. It is concluded that during this period, so has seen a shift trend towards of global search methods.

![Figure 2 Simulation-optimisation classification](image-url)
In order to increase efficiency SO techniques, several meta-heuristic approaches including GA, simulated annealing (SA) and tabu search (TS) have been explored. Several researches focused in particular on using GA search as an optimisation engine with simulation because of both simplicity and effectiveness. GA has been coupled with simulation to achieve different optimisations objectives. It has such advantages like independence of the application domain, suitability for very general optimisation problems, and robustness with respect to initial decisions. It deals with the random output of simulation experiments. It leaves local optima and finds the global one, and finally they need only a small amount of input information. Several of researches are focused in particular on utilising GA search as an optimisation engine with simulation because of both simplicity and effectiveness. For more detailed information, Wang et al. (2003) presented an object-oriented framework based on GA. This framework addressed many particular characteristics of green building design optimisation problems such as hierarchical variables and the coupling with simulation programmes. Azadivar and Tompkins (2003) presented the methodology which is a SO process where the qualitative variables and the structure of the system are the subjects of optimisation. The simulation models are automatically generated through an object-oriented process and are evaluated for various candidate configurations of the system. These candidates are suggested by a GA that automatically guides the system towards better solutions. Moreover, Pasandideh and Niaki (2006) employed simulation and GA to solve the multi-response statistical optimisation problem. Furthermore, Paul and Chaney (1998) showed an attempt to apply GA to the problem of optimising an existing Steelwork simulation model. A simple real-coded GA is presented and used to change the simulation model parameters. Fontanili et al. (2000) described the use of flow simulation and GA to optimise the management parameters in production. Authors develop two cases of optimisation to check the validity of the combination between flow simulation and a GA for obtaining of optimal parameters. Marseguerra et al. (2002) presented an optimisation approach based on the combination of a GA maximisation procedure with a Monte Carlo simulation. The approach is applied within the context of plant logistic management for what concerns the choice of maintenance and repair strategies. Azadivar and Wang (2000) presented an approach for solving facility layout optimisation problems for manufacturing systems with dynamic characteristics and qualitative and structural decision variables. The proposed approach integrates GA, computer simulation and an automated simulation model generator with a user-friendly interface. In another work, Marzouk and Moselhi (2004) presented a methodology for SO using GA and applies it to a newly developed simulation-based system for estimating the time and cost of earthmoving operations. The GA searches for a near-optimum that reduces project total cost and considers a set of qualitative and quantitative variables that influence earthmoving operations. In addition Boesel et al. (2003) described a new approach to SO has three distinct phases: the first phase allows the user to define the problem and input promising systems. The second phase generates new systems and the third phase uses a statistical procedure to determine which system is the best. In the search segment of the system-generation phase, we have incorporated adaptive error control so that our approach expends adequate but not excessive simulation effort to deal with sampling variability. Sudhir Ryan Daniel and Rajendran (2005) showed a GA where is proposed to optimise the base-stock levels with the objective of minimising the sum of holding and shortage costs in the entire SC. Simulation is used to evaluate the base-stock levels generated by the GA. The proposed GA is evaluated with the consideration of a variety of SC settings in order to test for its
Iterative simulation optimisation approach-based GA for LSP in MTO sector

roductness of performance across different SC scenarios. Moreover, Al-Aomar (2006) presented an approach to robust parameter design (PD) by optimising the response function of a simulation model that describes a production system exhibiting stochastic behaviour. The approach is based on GA search applied to the domain of a PD problem represented by a discrete event simulation (DES) model. The effective GA search has compensated for the numerous shortcomings of Taguchi’s experimental design with orthogonal arrays. The DES flexibility is used in the GA search to replace the explicit closed-form definition of the performance function. Furthermore, Suiadee and Tingsanchali (2007) combined simulation – GA optimisation model is developed to determine optimal reservoir operational rule curves of the Nam Oon Reservoir and Irrigation Project in Thailand. The GA and simulation models operate in parallel over time with interactions through their solution procedure. The combined simulation-GA model shows an excellent performance in terms of its optimisation results and efficient computation. In another work, Udhayakumar et al. (2011) introduced P-model CCDEA models, which include the concept of ‘satisficing’. The main contribution is the development of a stochastic simulation-based GA for solving CCDEA problems. Before this, they developed a complete framework of GA and developed a suitably designed Monte-Carlo simulation to process chance. Moreover, Ji et al. (2011) proposed a new simulation-based multi-objective genetic algorithm (SMOGA) approach to find a portfolio of reliable non dominant (Pareto) paths, a set of paths that is equally good or better at least in one objective space compared to all other paths, in stochastic networks while considering link travel time uncertainties and correlations among link travel times. Furthermore, Zhang et al. (2012) Proposed an automatic history matching method of reservoir numerical simulation based on improved GA the validity of which was proved by taking the SZ 36-1 typical reservoir of Bohai oilfield for example. Results show that the method proposed in this paper is effective for reservoir history matching problems due to its strong reliability and fast convergence speed. In another work, Yoo et al. (2012) introduced a genetic algorithm-based repetitive tasks simulation (GARTS) model for planning steel erection in high-rise building construction. This model produces an optimised movement path of a bolting robot for fabricating steel structures, proposes a collaboration plan between a robot and a worker, and quantifies the uncertainty of the duration of steel erection. As an illustrative case, the so-called robot-based construction automation system (RCA) was applied to a pilot project. The results showed the model’s capacities and justified its application to other extended types of robotic construction systems. In a recent study, Ammeri et al. (2013) developed a combined simulation and GA optimisation model is developed to solve the LSP in a MTO supply chain.

3 Illustrative case study

3.1 Model assumptions

The SC configuration investigated in this research is shown in Figure 3. A detailed of process description of the SC is a mandatory step for understanding what is implemented inside and how the simulation model works. The SC under study is composed of six locations (entities). Two locations, L1 and L2, serve a distribution or retail function and are exposed to independent customer order. Locations L3, L4, L5 and L6 serve a production function. These locations may be considered to be capacity constrained.
because time delay operations are assumed. The SC under study belongs to multi-stage category. Each finished product (P1, P2 and P3) has component items (manufacturing products) as indicated by the bill-of-distribution (BOD) and the bill-of-material (BOM).

Figure 4 describes BOD and BOM dictating the flow of material. Note that the BOM also shows the quantity of items required by each finished part. The number in brackets beside the part number shows the requirements for all finished part order. Part types P1 and P2 are both derived from P4. P3 is derived from P7. Since P7 is also a component of P4, the parts going directly to L2 could be considered spare or repair parts. At L3, P4 is assembled from three unit of P5, one unit of P7 and one P8 parts. At L5, P7 is produced from two units of P6. At L4, P8 is produced from one unit P6 and P5 is produced from one unit of raw material RM5. At L6, P6 is produced from one unit of raw material RM6. Supplies of RM5 and RM6 are assumed to be unlimited.

**Figure 3** Configuration of supply chain network
3.2 Model assumption

We consider a situation in which several types of parts are produced on the same SC. If the production is changed from one type to another, a set-up is needed. For some reasons, such as a large assortment of parts, which is subject to regular changes, a highly uncertain demand or unique parts, no safety stocks can be kept and we have to produce then according to customer’s specifications. For specificity, this research makes the following assumptions concerning the simulation parameters. For all part types, it was assumed that the required lot-size order quantity was not available from the upstream supplier before shipments could be released (i.e., lot splitting is allowed). Furthermore, for assembly operations it was assumed that the required lot-size quantities of all components were not also available before any components were released for shipment. A common transit time was then applied to all components so arrival at the assembly location was simultaneous. The transit times are shown alongside the arrows in Figure 4. Moreover, when there is a capacity constraints, lots arriving at manufacturing locations must undergo a setup time and a lot processing time. Mean setup and part processing times for processed and assembled parts are displayed in Table 1.

The lot setup times are stochastic and follow a normal distribution with a coefficient of variation of 0.3. The lot processing times are deterministic and based on multiplying the lot size times and the fixed part processing time. Processing of all lots in queue is based on FCFS. Transit times, also shown in Table 1, are defined as the time to move an available lot of inventory from an upstream location to a downstream location. The transit
times for all part types were assumed to be stochastic and follow a gamma distribution with a coefficient of variation of 0.1. No capacity constraints were assumed for inventory transportation.

Table 1  Supply chain data

<table>
<thead>
<tr>
<th>Part type</th>
<th>Mean setup time (hr)</th>
<th>Part processing time (hr)</th>
<th>Mean transit time (hr)</th>
<th>Lot size</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>0.5</td>
<td>0.009</td>
<td>10.0</td>
<td>200</td>
</tr>
<tr>
<td>P2</td>
<td>0.5</td>
<td>0.011</td>
<td>13.0</td>
<td>200</td>
</tr>
<tr>
<td>P3</td>
<td>0.9</td>
<td>0.023</td>
<td>26.0</td>
<td>300</td>
</tr>
<tr>
<td>P4</td>
<td>1.2</td>
<td>0.014</td>
<td>6.5</td>
<td>Unknowns to be optimised</td>
</tr>
<tr>
<td>P5</td>
<td>0.24</td>
<td>0.002</td>
<td>17.0</td>
<td></td>
</tr>
<tr>
<td>P6</td>
<td>0.50</td>
<td>0.003</td>
<td>15.0</td>
<td></td>
</tr>
<tr>
<td>P7</td>
<td>0.8</td>
<td>0.006</td>
<td>6.5</td>
<td></td>
</tr>
<tr>
<td>P8</td>
<td>1.6</td>
<td>0.007</td>
<td>6.5</td>
<td></td>
</tr>
</tbody>
</table>

Furthermore, the customer order which consists of fixed lot size of the same product type follows a gamma distribution with a mean of 4,000 units per week. One week is assumed to be equal to five working days. Times are given in hours, using the assumption there are 40 hours per week (or eight hours per day). Daily order variation is determined on the basis of having a week order coefficient of variation of 0.1. Based on customer requirements and specifications, lot sizes for product types P1, P2 and P3 are compulsory fixed at 200 units, 200 units, and 300 units, respectively.

3.3 Performance measure

Many performance measures (metrics) can be considered for SC analysis such as work-in-process, mean tardiness, mean flow time and others. The coherence between the criteria of performance guarantees the overall performance of the SC. In MTO production strategy, the main priority is to minimise mean lot tardiness in order to avoid associated penalties. On the other hand, it is also important to minimise mean lot earliness and its related extra storage costs. For this reason, the most appropriate performance measure would be the OMFT. There is a fixed target value (delivery promise date) proposed for each customer order. This target value must be framed between the lower and the upper values which are fixed for reduce cost as indicated in Table 2.

Table 2  Order supply chain objectives

<table>
<thead>
<tr>
<th>Performance measure</th>
<th>Lower (hr)</th>
<th>Target (hr)</th>
<th>Upper (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMFT P1</td>
<td>57</td>
<td>60</td>
<td>63</td>
</tr>
<tr>
<td>OMFT P2</td>
<td>62</td>
<td>65</td>
<td>68</td>
</tr>
<tr>
<td>OMFT P3</td>
<td>57</td>
<td>60</td>
<td>63</td>
</tr>
</tbody>
</table>
4 The adopted approach

4.1 Methodology description

Recent developments in the area of optimisation have allowed the creation of intelligent search methods capable of finding optimal or near optimal solutions to complex problems. The simulation and optimisation steps alternate with each other. For any decision suggested by the optimiser, information about the decisions performance is collected through simulation. Optimisation and simulation push each other as long as we have not found an acceptable decision. The procedure explained here is the combination of two processes. First, a set of configurations for the system is selected to form the original population for GA. Each configuration in this set consists of a choice of a machine in each station, a choice of dispatching rule, and a choice of other characteristics of the components of the system. These choices are made from the information provided by the user on the available alternatives and are represented in the form of a string. A simulation model generator has been developed that can translate the input given in the form of a string into a simulation model. Through this, each configuration is translated into a simulation model. Each simulation model is run and the response is recorded as the fitness value for that configuration. These values in turn are provided to the GA to generate the next population through the designated procedure. The information on the members of this new population is then provided to the model generator so that new simulation models can be built and evaluated. This interaction continues until either there is no significant difference among the responses of all members of a generation or a certain level of budget for simulation effort has reached. The structure of the simulation-based optimisation framework is shown in Figure 5.

Figure 5 Simulation-based optimisation framework

<table>
<thead>
<tr>
<th>Genetic algorithm</th>
<th>Performance evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candidate solution</td>
<td>Simulation model</td>
</tr>
</tbody>
</table>

4.2 The adopted model

The procedure explained here is the combination of two processes. First, a set of configurations for the system is selected to form the original population for GA. The GA provides the initial solutions to the simulation model, thus we can obtain the system performances of each initial solution. Then, the simulation model sends these performances values to the GA. The GA can compute several offspring solutions (the elitist of current solutions are always kept) and these solutions are considered as the input of the simulation model for the next iteration. At the same time, we improve the crossover probability and the mutation probability each consecutive iterations. This process is repeated until the stopping criterion is satisfied.

The VBA tool is integrated in the ARENA environment. It provides a link between the simulation model and the optimisation algorithm. The GA starts immediately after the generation of the initial population. This step is carried out in parallel during the
execution of the simulation model. The simulation model represents the implementation of the fitness function of the GA, needed to evaluate each member (chromosome) of the population. Figure 6 illustrates the potential range of data exchange available to the model developer using VBA code. This tutorial concentrates on the exchange of data between Arena and Excel.

**Figure 6** Adopted method using genetic algorithm

To demonstrate and validate the methodology developed in this paper, a case study which is adopted from literature, deals with a LSP in MTO supply chain.

### 4.3 Simulation model

The model of the studied LSP is realised with the ARENA simulation environment. The global flowchart description is demonstrated in Figure 7. The main portion of the model’s operation consists of logic modules to represent order arrivals, SC locations and order delivery. For instance, the module ‘create P1 order’ ensures the entry of the product P1 lots within the system. Then each lot assigned a set of attributes such as part type and sequence routing, via the module ‘assign P1 attributes’. Additionally, as the part proceeds through the SC, different attributes record the time delays associated with material handling, setup and process time. All locations can be modelled by a set of machines, in which each one is modelled using the ‘enter-process-leave’ module sequence. Finally, each order leaves the SC through the module ‘dispose’.
4.4 GA representation

The first stage of the GA process involves encoding information. In the LSP, each gene represents a lot size for each manufacturing product. The length of the chromosomes represents the number of products considered in the LSP. A typical chromosome, shown in Figure 8.

![Figure 8 - A typical chromosome encoding](image)

The second step in GA is to initialise the population of chromosomes. Crossover combines the characteristics of two parents to produce offspring (Figure 9), whereas mutation produces random changes in a single chromosome (Figure 10). As shown in Figure 9, the initial population was randomly chosen between 100 and 600 individuals. The goal of this research is to determine a fixed optimal lot size for each manufacturing product type that will ensure OMFT target value for each finished product type. For this purpose, the objective function for LSP is to minimise the sum of the differences between OMFT targets.

Chromosomes are randomly selected for crossover and mutation operations with the probabilities specified. A crossover operator called single point with repairs is designed. This operator first chooses a point of one parent and inserting the segment to the left of this into the first offspring in the same order. The remaining positions are filled from the other parent, select the gene from left to right of the chromosome following the rule that gene cannot be identical. Figure 9 shows an example where new solutions are created by crossover.

![Figure 9 - Single-point crossover](image)
After the crossover operation is performed, mutation occurs. This prevents all solutions in the population from falling into a local optimum of solved problems. In this study, individual genes of new offspring are changed randomly with probability 1%.

In every crossover and mutation, new candidate solutions are created. The new solutions may be better than the previous. This means that performing solutions are to be replaced using a replacement strategy. Several strategies have been suggested in the past. These include the following: probabilistic replacement, crowding strategy and elitist strategy. In this study, the elitist strategy has been adopted. All steps of the GA have been coded using Microsoft VBA.

4.5 VBA language

VBA is an event-driven programming language which was first introduced by Microsoft in 1993 to give Excel 5.0 a more robust OO language for writing macros and automating the use of Excel. Actually, each Office application supports VBA (Seppanen, 2000; Getz and Gilbert, 1997). The user interface is identical in all these applications that support VBA.

Pressing the Alt-F11 key on any of these applications loads the editor illustrated in Figure 11. The structure of the VBA language is beyond the scope of this presentation but documented in numerous texts, such as in Seppanen (2000). The new VBA user has much to learn but help is always as close as the F1 and F2 keys. Pressing the F2 key in the Microsoft Visual Basic editor provides lists of available constants, functions, and properties. From these three constructs, only the value of a property can be changed and then only if it has not been declared as read only. Actually, VBA is quite easy to learn because the built-in editor automatically checks the syntax of each code line as it is entered. The Debug compile feature checks that all variables have been defined prior to attempting execution. Finally even during model execution, many VBA errors can be interactively debugged and corrected without restarting the execution of the arena model.

Figure 10 Random single case mutation

After the crossover operation is performed, mutation occurs. This prevents all solutions in the population from falling into a local optimum of solved problems. In this study, individual genes of new offspring are changed randomly with probability 1%.

In every crossover and mutation, new candidate solutions are created. The new solutions may be better than the previous. This means that performing solutions are to be replaced using a replacement strategy. Several strategies have been suggested in the past. These include the following: probabilistic replacement, crowding strategy and elitist strategy. In this study, the elitist strategy has been adopted. All steps of the GA have been coded using Microsoft VBA.

4.5 VBA language

VBA is an event-driven programming language which was first introduced by Microsoft in 1993 to give Excel 5.0 a more robust OO language for writing macros and automating the use of Excel. Actually, each Office application supports VBA (Seppanen, 2000; Getz and Gilbert, 1997). The user interface is identical in all these applications that support VBA.

Pressing the Alt-F11 key on any of these applications loads the editor illustrated in Figure 11. The structure of the VBA language is beyond the scope of this presentation but documented in numerous texts, such as in Seppanen (2000). The new VBA user has much to learn but help is always as close as the F1 and F2 keys. Pressing the F2 key in the Microsoft Visual Basic editor provides lists of available constants, functions, and properties. From these three constructs, only the value of a property can be changed and then only if it has not been declared as read only. Actually, VBA is quite easy to learn because the built-in editor automatically checks the syntax of each code line as it is entered. The Debug compile feature checks that all variables have been defined prior to attempting execution. Finally even during model execution, many VBA errors can be interactively debugged and corrected without restarting the execution of the arena model.

Figure 11 Example of the VBA Editor Window (see online version for colours)
The VBA tool is integrated in the ARENA environment. It provides a link between the simulation model and the optimisation algorithm. The GA starts immediately after the generation of the initial population. This step is carried out in parallel during the execution of the simulation model. The simulation model represents the implementation of the fitness function of the GA needed to evaluate each member (chromosome) of the population.

5 The adopted approach simulation results and comparison

The GA and simulation models operate in parallel over time with interactions. GA model is implemented using VBA language. The evolutionary environment for this GA experiment is given as follows. The population size is 30, the mutation rate is 0.01 and the maximum number of generations allowed is 20.

For the same case study, Ammeri et al. (2011) proposed an approach that suggests using heuristic search used in OptQuest. They obtain in optimal configuration that: lot size of P4 would be set at 479.5 units. Lot size of P5 would be set at 438 units, lot size of P6 would be set at 500 units, lot size of P7 would be set at 359 units and lot size of P8 would be set at 147 units. Table 2 summarises the optimal value of OMFT performance found in these approaches. The Figure 12 describes the evolution of the quality of solutions throughout the iterations and shows that the normal application of the AG indicates that the optimal solution is reached at the 20th iteration as the ideal optimal solution (fitness = 0) is not reached.

Figure 12  Evolution of the quality of solutions throughout the iterations

As shown in Table 3, the case study problem is to determine the fixed lot size for each manufacturing product type P1, P2 and P3, which will ensure OMFT target. The optimal configuration of 20 iterations is indicated in Table 3. The results of the comparative study between methods show that all methods give an acceptable solution. However the results obtained with the GA associated with the flow simulation are very promising because they are better than all the results that we have found. This result can be justified that the GA tends to escape more easily from local optima due to its population-based nature. In addition, it does not require specific domain information although it can exploit it if such information is available.
Table 3 Comparative results

<table>
<thead>
<tr>
<th></th>
<th>Global search method: OptQuest (hr)</th>
<th>Simulation-based GA (hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P4</td>
<td>479</td>
<td>357</td>
</tr>
<tr>
<td>P5</td>
<td>437</td>
<td>422</td>
</tr>
<tr>
<td>P6</td>
<td>500</td>
<td>153</td>
</tr>
<tr>
<td>P7</td>
<td>358</td>
<td>347</td>
</tr>
<tr>
<td>P8</td>
<td>146</td>
<td>532</td>
</tr>
<tr>
<td>OMFTP1</td>
<td>61.30</td>
<td>60.12</td>
</tr>
<tr>
<td>OMFTP2</td>
<td>63.95</td>
<td>64.10</td>
</tr>
<tr>
<td>OMFTP3</td>
<td>57.00</td>
<td>59.64</td>
</tr>
<tr>
<td>Fitness function</td>
<td>5.35</td>
<td>1.38</td>
</tr>
</tbody>
</table>

6 Conclusions

In this paper we propose a hybrid simulation optimisation approach that combines both the optimisation and the simulation modelling approaches. This study integrates simulation techniques with a GA using an iterative process. GA model is implemented using VBA language. To demonstrate the feasibility of this approach, the hybrid approach is applied to LSP in MTO supply chain. The SC under study, which operates in MTO environment (no possibility for stock keeping and limited production capacity), is characterised by multi-product, multi-stage, multi-location production planning with capacity-constrained and stochastic parameters such as lot arrivals order, transit time, setup time, processing time, etc. For this reason, we compared the different outcomes and we proved that our method using simulation techniques with a genetic algorithm gives a good satisfactory solution. It should be noted that the motivating case study is just a notional example.

References


A. Ammeri et al.


