First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Building a decision support system for long-term production planning using a distributed genetic algorithm

Florin Stoica

Abstract

This paper represents an approach of using a distributed genetic algorithm for generating an aggregative production plan to maximize the total profit of the firm. The described methodology provides a tool which assists the company management in finding an optimal long-term production plan. The entire system is composed by a business information system - a front-end for the database of the ERP system, a simulation model and a distributed genetic algorithm. The purpose of the integrated system is to help operative management personnel to take decisions with respect to long-term production planning. In order to improve the response time, the entire system is distributed across several network machines.

1 Introduction

Most real life optimization and scheduling problems are too complex to be solved completely. The complexity of real life problems often exceeds the ability of classic methods. In such cases decision-makers prepare and execute a set of scenarios on the simulation model and hope that at least one scenario will be good enough to be used as a production plan [15].

A long time goal for scheduling optimization research has been to find an approach that will lead to qualitative solutions in a relatively short computational time. The development of decision-making methodologies is currently headed in the direction of integration of simulation and search algorithms. This leads to a new approach, which successfully joins simulation and optimization. The proposed approach supports man-machine interaction in operational and long-term production planning [16].

A group of widely known meta-heuristic search algorithms are genetic algorithms (GA). Evolutionary algorithms are a very effective tool that enables solving complicated practical optimization problems [2]. An important characteristic of evolutionary algorithms is their simplicity and versatility. The principal advantage of GAs is their inherent ability to intelligently explore the solution space from many different points simultaneously enabling higher probability for locating global optimum without having to analyze all possible solutions available and without requiring derivatives (or numerical approximations) or other auxiliary knowledge. Their main drawback is a long calculation time. A solution to this problem is parallelizing genetic algorithms.

With computer imitation of simplified and idealized evolution, an individual solution-chromosome represents a possible solution to our problem. Chromosome fitness is calculated with a fitness function. After being evaluated with a fitness function, each chromosome in population receives its fitness value.

The optimization is based on a genetic algorithm which uses a new co-mutation operator called LR-Mijn, capable of operating on a set of adjacent bits in one single step.

In present, there is a major interest in design of powerful mutation operators, in order to solve practical problems which can not be efficiently resolved using standard genetic operators. These new operators are called co-mutation operators. In [7] was presented a co-mutation operator called Mijn, capable of operating on a set of adjacent bits in one single step. In [12] we introduced and studied a new co-mutation operator which we denoted by LR-Mijn and we proved that it offers superior performances than Mijn operator.

The paper is organized as follows. In Section 2 we make a brief presentation of the architecture of our decision-support system, called DGA-SIM. We also present the basic idea of evolutionary method adopted for optimization process. In section 3 is introduced the co-mutation operator LR-Mijn. The evolutionary algorithm based on the LR-Mijn operator, used within the decision-support system is presented in section 4. Section 5 contains the description of the simulation model and its integration in the DGA-SIM system. Our solution for parallelizing the genetic algorithm is also presented in section 5. Conclusions and further directions of study can be found in section 6.

2 The architecture of the decision-support system

Companies need to be flexible to compete for the market share and adapt to market demands by offering competitive prices and quality. This calls for a wide assortment of products or product types, small production costs, high productivity etc. Simulation is a strong interactive tool that helps decision-makers improves the efficiency of enterprise actions. The ability of simulation to show a real process on the computer with the consideration of uncertainty is a big advantage when analyzing system behavior in complex situations [15].

Our system is composed by a business information system - a front-end for the database of the ERP system, a simulation model and a distributed genetic algorithm, and is called in the following the DGA-SIM system.

A simulation model will be used for fitness function computation of genetic algorithm results, as well as for visual representation of qualitative evaluation of a chosen production plan following genetic algorithm optimization.

The value of the fitness function is computed by the simulation model, which uses the respective chromosome as input data. The value returned from the simulation model is used to evaluate that chromosome, which encodes a possible production plan.

By applying the genetic operations on the members of a population of decisional rules at the moment t, it results a new population of rules that shall be used at the moment t+1. The population from the initial moment, $t = 0$ is generated randomly and the genetic operations are applied iteratively until the moment T (at which the condition for stopping the algorithm is accomplished) [16].

The above iterative process may be interpreted economically as it follows. The evolution process has as objective the finding of the successful individuals. The binary strings of these individuals (chromosomes) with high fitness values (a high profit) will be the basis for building a new generation (population). The strings with lower fitness values, which represent decisions to produce with a low profit, find few successors (or none) in the following generation.

The purpose of the integrated system is to aid operative management personnel in production planning and marketing strategies (incorporated in simulation model). The main advantage of the presented system is to enhance man-machine interaction in production planning, since the computer is able to produce several acceptable solutions using the given data and a set of criteria [16].

After completion of the optimization process, the most suitable production plans are simulated on the visual model of the system. Using chosen parameters and according to defined criteria, the decision-maker is motivated to search for results which will have the most advantageous influence on the whole production process. The result, which is selected after simulation on the visual simulation model, becomes the proposed production plan. Then the user modifies it, if necessary.

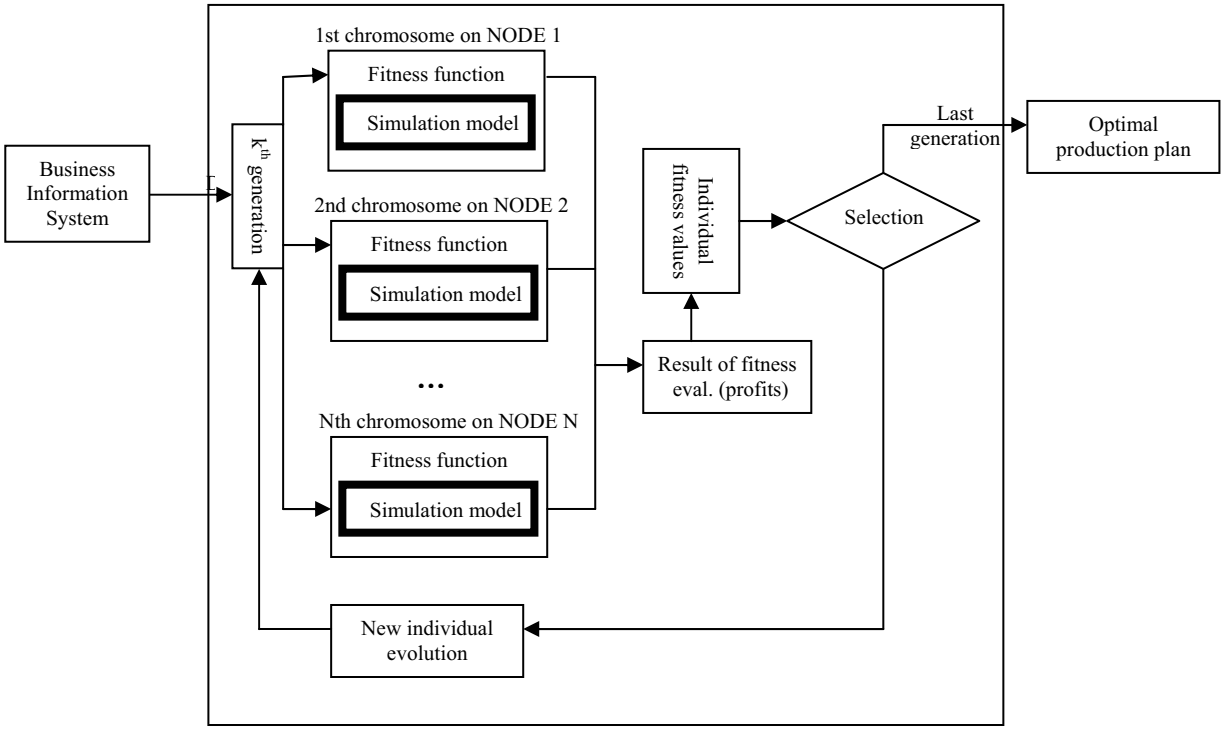

Fig. 1 The architecture of the decision-support system

Every chromosome codes a possible production plan. The quality of a chromosome is represented by the amount of profit which result from the simulation model if it receives as input data the production plan coded in that chromosome.

Parallelizing genetic algorithm is easy to be implemented by distributing one chromosome to one computer such that simulation model with different possible production plans can be run simultaneously at different computers.

The user interface of the simulation model of the DGA-SIM system is based on Microsoft EXCEL. The calculations for a specific production plan coded within a chromosome are performed in EXCEL, and result is returned in the GA as the evaluation of the respective production plan (the fitness value).

3 The LR-M_{ijn} operator

In this section we define the co-mutation operator called LR-M_{ijn}. Our LR-M_{ijn} operator finds the longest sequence of σ_p elements, situated *in the left or in the right* of the position p. If the longest sequence is in the left of p, the LR-M_{ijn} behaves as M_{ijn}, otherwise the LR-M_{ijn} will operate on the set of bits starting from p and going to the *right*.

Let us consider a generic alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_s\}$ composed by s ≥2 different symbols. The set of all sequences of length *l* over the alphabet $\mathcal{A}$ will be denoted with $\Sigma = \mathcal{A}^l$.

In the following we shall denote with σ a generic string, and $\sigma = \sigma_{l-1}\ldots\sigma_0 \in \Sigma = \mathcal{A}^l$, where $\sigma_q \in \mathcal{A}$ $\forall$ q ∈ {0, …, *l*-1}. Through σ(q,i) we denote that on position q within the sequence σ there is the symbol a_i of the alphabet $\mathcal{A}$., $\sigma^z_{p,j}$ denotes the presence of z symbols a_j within the sequence σ, starting from the position p and going left and $\underset{left,m}{\overset{right,n}{\sigma(p,i)}}$ specify the presence of symbol a_i on position p within

the sequence σ, between *right* symbols a_n on the right and *left* symbols a_m on the left. We suppose that $\sigma = \sigma(l\text{-}1)\ldots \sigma(p+left+1,m)\ \overset{right,i}{\underset{left,i}{\sigma(p,i)}}\ \sigma(p\text{-}right\text{-}1,n)\ldots \sigma(0)$.

The M_{ijn} operator is the mutation operator defined in [7]:

$M_{ijn} : \sigma \in \Sigma, p \in \{0, \ldots, l\text{-}1\} \rightarrow \sigma' \in \Sigma' \subset \Sigma$, where p is randomly chosen

$$\sigma = \sigma_{l-1}\ldots\sigma_{p+n}\sigma_{p+n-1,i}\,\sigma_{p,j}^{n-1}\,\sigma_{p-1}\ldots\sigma_0 \xrightarrow{M_{ijn}} \sigma' = \sigma_{l-1}\ldots\sigma_{p+n}\sigma_{p+n-1,j}\,\sigma_{p,i}^{n-1}\,\sigma_{p-1}\ldots\sigma_0, \quad \text{(i)}$$

for $n < l - p + 1$ and

$$(\sigma = \sigma_{p,j}^{l-p}\,\sigma_{p-1}\ldots\sigma_0 \xrightarrow{M_{ijn}} \sigma' = \sigma_{p,k}^{l-p}\,\sigma_{p-1}\ldots\sigma_0, \quad \text{(ii)}$$

for $n = l - p + 1$, with $a_k \neq a_j$ randomly chosen in $\mathcal{A}$.

In [12], we introduced and study the properties of LR-M_{ijn} co-mutation operator.

Definition 1 *Formally, the LR-M_{ijn} operator is defined as follows*:

(i) If $p \neq right$ and $p \neq l - left - 1$,

$$\text{LR-Mijn}(\sigma) = \begin{cases} \sigma_{left} = \sigma(l-1)\ldots\sigma(p+left+1,i)\ \overset{right,i}{\underset{left,m}{\sigma(p,m)}} \\ \qquad \sigma(p-right-1,n)\ldots\sigma(0) \quad for\ left > right \\ \sigma_{right} = \sigma(l-1)\ldots\sigma(p+left+1,m)\ \overset{right,n}{\underset{left,i}{\sigma(p,n)}} \\ \qquad \sigma(p-right-1,i)\ldots\sigma(0) \quad for\ left < right \\ \sigma_{right}\ or\ \sigma_{left}\ \ for\ left = right,\ with\ probability\ 0.5 \end{cases}$$

(ii) If $p = right$ and $p \neq l - left - 1$,

$\sigma = \sigma(l\text{-}1)\ldots \sigma(p+left+1,m)\ \overset{right,i}{\underset{left,i}{\sigma(p,i)}}$ and $\text{LR-Mijn}(\sigma) = \sigma(l\text{-}1)\ldots \sigma(p+left+1,m)\ \overset{right,k}{\underset{left,i}{\sigma(p,k)}}$,

where $k \neq i$ (randomly chosen).

(iii) If $p \neq right$ and $p = l - left - 1$,

$\sigma = \overset{right,i}{\underset{left,i}{\sigma(p,i)}}\ \sigma(p\text{-}right\text{-}1,n)\ldots \sigma(0)$ and

$\text{LR-Mijn}(\sigma) = \overset{right,i}{\underset{left,k}{\sigma(p,k)}}\ \sigma(p\text{-}right\text{-}1,n)\ldots \sigma(0)$, where $k \neq i$ (randomly chosen).

(iv) If $p = right$ and $p = l - left - 1$, $\sigma = \overset{right,i}{\underset{left,i}{\sigma(p,i)}}$

$$\text{LR-Mijn}(\sigma) = \begin{cases} \sigma_{left} = \overset{right,i}{\underset{left,k}{\sigma(p,k)}},\ for\ left > right,\ where\ \ k \neq i \\ \sigma_{right} = \overset{right,k}{\underset{left,i}{\sigma(p,k)}},\ for\ left < right,\ where\ \ k \neq i \\ \sigma_{right}\ or\ \sigma_{left}\ \ for\ left = right,\ with\ probability\ 0.5 \end{cases}$$

As an example, let us consider the binary case, the string σ = 11110000 and the randomly chosen application point p = 2. In this case, $\sigma_2 = 0$, so we have to find the longest sequence of 0 within string σ, starting from position p. This sequence goes to the right, and because we have reached the end of

the string, and no occurrence of 1 has been met, the new string obtained after the application of LR-M_{ijn} is 11110111.

The commutation operator LR-M_{ijn} allows long jumps [8], thus the search can reach very far points from where the search currently is. We proved in [12] that the LR-M_{ijn} operator performs more long jumps than M_{ijn}, which leads to a better convergence of an evolutionary algorithm based on the LR-M_{ijn} in comparison with an algorithm based on the M_{ijn} operator.

In the following implementation, we will consider that $\mathcal{A}$ is the binary alphabet, $\mathcal{A} = \{0, 1\}$.

4 The evolutionary algorithm based on LR-M_{ijn} operator

The basic scheme for our algorithm, called in the following LR-MEA, is described as follows:

```
Procedure LR-MEA
 begin
  t = 0
  Initialize randomly population P(t) with P elements;
  Evaluate P (t) by using fitness function;
  while not Terminated
    for j = 1 to P-1 do
      - select randomly one element among the best T% from P(t);
      - mutate it using LR-Mijn;
      - evaluate the obtained offspring;
      - insert it into P'(t).
    end for
    Choose the best element from P(t) and Insert it into P'(t)
    P(t+1) = P'(t)
    t = t + 1
  end while
 end
```

5 The simulation model

The simulation model is implemented as an Excel application. It is based on real data provided by the Business Information System but also on forecasted data (e.g. sales quantities and values for the following months). The model take account of many dates and variables: sales (values and quantities), raw material costs, discounts, packaging costs, direct & indirect labor costs, energy cost, depreciation from the rate of exchange, warehousing and logistic costs, transport costs, advertising & promotions, etc. The main workbook contains a few sheets, a VBA module, and results are synthesized in the pivot table described in Figure 2.

The role of the simulation model in the genetic algorithm is presented in the Figure 3.

In fact, the simulation model represents the implementation of the fitness function of the genetic algorithm, needed in the procedure LR-MEA to evaluate each member (chromosome) of the population.

The probability to select a certain chromosome (possible production plan) in the next generation is related with its performance in simulated conditions (the profit provided by the simulation model). That profit determines the fitness value of the respective possible solution of our optimization problem.

Because the DGA-SIM system is implemented in Java as main language, was necessary a bridge between Java code and the simulation model, implemented in Excel. Our choice for this purpose was JExcelAPI (http://jexcelapi.sourceforge.net/), a mature, open source java API enabling developers to read, write, and modify Excel spreadsheets dynamically [14].

Parallelizing genetic algorithm was implemented by distributing one chromosome to one computer such that simulation model with different possible production plans can be run simultaneously at different computers.

	A	B
1		Grand Total
2		
3	Values	
4	Kg	12,567,833
5	Sku	140,244,000
6	Sales	328,992,355
7	Discounts %	10.50%
8	Turnover	190,458,000
9	Total Costs	135,200,278
10	Direct Labour	3,189,330
11	...	...
12	Directs Costs	82,000,388
13	Energy gas and water	2,000,477
14	Indirect Labour	6,300,212
15	Depreciation	1,903,235
16	Warehousing	1,589,345
17	Logistic platforms	560,233
18	Advertising & Promotions	9,280,023
19	Advertising & Promotions %	5.05%
20	Overheads	14,934,770
21	Overheads %	8.93%
22	Profit	8,950,168

Fig. 2 A pivot table from the simulation model [16]

The implementation of the distributed genetic algorithm is based on Java Remote Method Invocation (RMI). Java RMI enables the programmer to create distributed Java technology-based to Java technology-based applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, hosted by different computers. RMI provides a standard remote communication between programs written in the Java programming language [17].

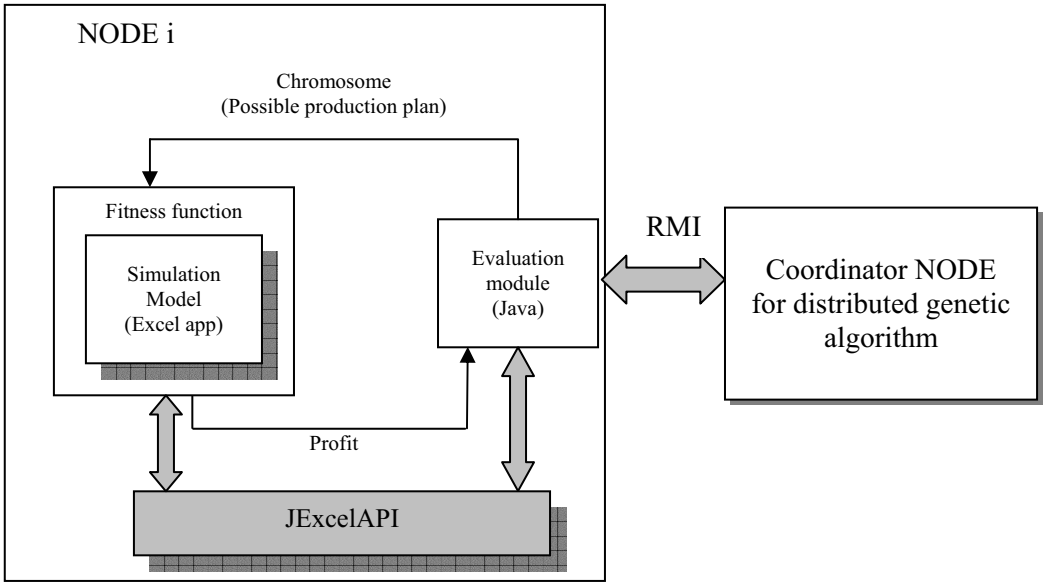

Fig. 3 Implementation of the fitness function through the simulation model

The parallel code runs in a network of 20 computers to demonstrate the efficiency of the distributed genetic algorithm to our problem. Therefore we are using 20 chromosomes (because of 20 network computers) for 20 production plans simulated/evaluated in parallel in one generation.

The role of the coordinator node is to centralize the results (profits) provided by each individual node, in order to evaluate in the following step the whole population. Then, the best production plans (chromosomes) are selected for the next generation.

6 Conclusions and further directions of study

The purpose of the DGA-SIM integrated system is to aid operative management personnel in production planning and marketing strategies (incorporated in simulation model). The main advantage of the presented system is to enhance man-machine interaction in production planning, since the computer is able to produce several acceptable schedules using the given data and a set of criteria.

The DGA-SIM system is currently under evaluation in a big company from Sibiu, Romania, which was interested in its acquisition.

As a further direction of study we want to compare the results obtained by using different genetic operators and to evaluate real codifications of variables, instead of current binary one.

Another improvement of our study is represented by designing of appropriate simulation models for multi-criteria optimization with genetic algorithms.

Acknowledgement: This paper is founded from the ULBS research project 164/2009.

References

[1] Jaber A. Q, Hidehiko Y., F., Ramli R., Machine Learning in Production Systems Design Using Genetic Algorithms, *International Journal of Computational Intelligence*, No. 4, 2008, pp. 72-79.

[2] Kofjač D., Kljajić M., Application of Genetic Algorithms and Visual Simulation in a Real-Case Production Optimization, *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, Issue 12, Volume 3, December 2008, pp. 992-1001.

[3] Radhakrishnan P., Prasad V. M., Gopalan M.R., Optimizing Inventory Using Genetic Algorithm for Efficient Supply Chain Management, *Journal of Computer Science 5 (3)*, 2009, pp. 233 - 241.

[4] Lo Chih-Yao, Advance of Dynamic Production-Inventory Strategy for Multiple Policies Using Genetic Algorithm, *Information Technology Journal 7 (4)*, 2008, pp. 647-653.

[5] De Falco I., An introduction to Evolutionary Algorithms and their application to the Aerofoil Design Problem – *Part I: the Algorithms*, *von Karman Lecture Series on Fluid Dynamics*, Bruxelles, April 1997

[6] De Falco I., Del Balio R, Della Cioppa A., Tarantino E., A Comparative Analysis of Evolutionary Algorithms for Function Optimisation, *Research Institute on Parallel Information Systems*, National Research Council of Italy, 1998

[7] De Falco I, A. Iazzetta, A. Della Cioppa, Tarantino E., The Effectiveness of Co-mutation in Evolutionary Algorithms: the M_{ijn} operator, *Research Institute on Parallel Information Systems*, National Research Council of Italy, 2000

[8] De Falco I., Iazzetta A., Della Cioppa A., Tarantino E., M_{ijn} Mutation Operator for Aerofoil Design Optimisation, *Research Institute on Parallel Information Systems*, National Research Council of Italy, 2001

[9] Chen J., Using Genetic Algorithms to Solve a Production-Inventory Model, *International Journal of Business and Management*, Vol. 2, No. 2, 2007, pp. 38-41.

[10] Krishnakumar K., Goldberg D., Control system optimization using genetic algorithm, *Journal of Guidance, Control, and Dynamics*, no. 15(3), 1992, pp. 735-740.

[11] Zhu X., Huang Y., Doyle J., Genetic algorithms and simulated annealing for robustness analysis, *Proceedings of the American Control Conference*, Albuquerque, New Mexico, 1997, pp. 3756-3760.

[12] Stoica F., Simian D., Simian C., *A new co-mutation genetic operator*, Proceedings of the 9th WSEAS International Conference on Evolutionary Computing, Sofia, Bulgaria, May 2008, pp. 76-81.

[13] Vapnik V., *The Nature of Statistical Learning Theory*, Springer Verlag, 1995.

[14] Java Excel API, http://jexcelapi.sourceforge.net

[15] Kljajić M., Bernik I., Škraba A., Leskovar R., Integral simulation approach to decision assessment in enterprises, *Shaping future with simulation: proceedings of the 4th International Eurosim 2001 Congress*, Delft University of Technology, 2001.

[16]Stoica F., Cacovean L., Using genetic algorithms and simulation as decision support in marketing strategies and long-term production planning, *Proceedings of the 9th WSEAS International Conference on SIMULATION, MODELLING and OPTIMIZATION*, Budapest, Hungary, September 3-5, 2009
[17]Remote Method Invocation Home, http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp

Florin Stoica
University "Lucian Blaga" of Sibiu
Faculty of Sciences
Computer Science Department
5-7 I. Ratiu str., Sibiu, 550021
ROMANIA
E-mail: *florin.stoica@ulbsibiu.ro*