# DEVELOPMENT OF MULTI-OBJECTIVE SIMULATION-BASED GENETIC ALGORITHM FOR SUPPLY CHAIN CYCLIC PLANNING AND OPTIMISATION

## Galina Merkuryeva, Liana Napalkova

*Riga Technical University,*
*1 Kalku Street, LV-1658 Riga, Latvia*
*E-mail: gm@itl.rtu.lv, liana@itl.rtu.lv*

**Abstract:** This paper develops a multi-objective simulation-based genetic algorithm (MOSGA) for multi-echelon supply chain cyclic planning and optimisation. The problem involves a search in high dimensional space with different ranges for decision variables scales, multiple objectives and problem specific constraints, such as power-of-two and nested/inverted-nested planning policies. In order to find the optimal solution, different parameters of genetic algorithm including the population sizing, crossover and mutation probabilities, selection and reproduction strategies and convergence criteria are investigated. For finding approximations of the Pareto optimal set, the non-dominated sorting approach is used.

**Keywords:** multi-echelon cyclic planning, multi-objective optimisation, genetic algorithms.

## 1. Introduction

For the last years, there has been increasing attention placed on the performance, design, and analysis of multi-echelon supply chains (Merkuryeva, Napalkova, 2007). There are two different approaches used to manage supply chains: *single-echelon* and *multi-echelon approaches*.

The single-echelon approach splits multi-level supply chain into separate stages providing suboptimal solutions. In contrast to it, the multi-echelon approach, which could be applied to non-cyclic and cyclic policies, considers managing all the echelons in a holistic way and, thus, optimises the global supply chain performance.

Compared to non-cyclic policies, which are more preferable from theoretical point of view, *cyclic planning* in multi-echelon environment has more practical benefits, because it provides easy control, reduced administrative costs and safety stocks, and elimination of bullwhip effect. The main idea of cyclic planning is to use cyclic schedules at each echelon and synchronize them with one another (Merkuryev, Merkuryeva *et al.*, 2007).

Multi-echelon cyclic planning and optimisation refers to the class of multi-objective optimization problems, which are usually characterized by a large search space of decision variables, conflicting and stochastic objectives etc. (Chen, 2003). While there is no a single optimal solution for a number of conflicting objectives, the development of an algorithm, which gives a large number of alternative solutions lying near the Pareto-optimal frontier and tackles the variations of a response generated from the uncertainties in the decision variables and/or parameters, is of great practical value.

The motivation for the current study is to develop the multi-objective genetic algorithm for solving this optimisation problem in accordance with research conducted within ECLIPS Specific Targeted Research Project of the European Commission "Extended Collaborative Integrated Life Cycle Supply Chain Planning System".

This paper is divided into eight sections. Section 2 describes multiple objectives and the optimisation problem statement. Section 3 contains requirements imposed on the development of genetic algorithm (GA) to solve the optimisation problem. Section 4 provides arguments in favour of the development of the multi-objective simulation-based GA. The main features of the used GA are discussed in sections 5.1-5.8. Section 6 presents a prototype of that algorithm. Conclusions are represented in Section 7.

## 2. Optimisation problem statement

The multi-objective simulation-optimisation problem can be symbolically represented in compact form as:

$$\text{Minimize } E[\boldsymbol{y}] = E[F(\boldsymbol{x})] = E[(f_1(\boldsymbol{x}),...,f_n(\boldsymbol{x}))],$$
$$\text{where } \boldsymbol{x} = (x_1,...,x_m) \in X, \quad \boldsymbol{y} = (y_1,...,y_n) \in Y, \tag{1}$$

where $E[\cdot]$ is a mathematical expectation, $\boldsymbol{x}$ is called a decision variable vector; $\boldsymbol{y}$ is called an objective vector; $x_1,...x_m$ denote $m$ decision variables; $y_1,...,y_n$ denote $n$ multiple objectives; $X$ is a decision variable space; $Y$ is an objective space.

Proceeding from (1), the solution of multi-objective optimisation problem is a vector of decision variables $\boldsymbol{x}$ that satisfies all feasible constraints and provides the best trade-off between multiple objectives. For finding trade-off solutions, it is necessary to use the principles of dominance and Pareto optimality.

The trade-off solution $\boldsymbol{x} \in X$ is said to be Pareto optimal with respect to $X$ if and only if there is no $\boldsymbol{x}' \in X$, for which $F(\boldsymbol{x}') = (f_1(\boldsymbol{x}'),...,f_n(\boldsymbol{x}'))$ dominates $F(\boldsymbol{x}) = (f_1(\boldsymbol{x}),...,f_n(\boldsymbol{x}))$. At that, the set of objective vectors $F(\boldsymbol{x})$ corresponding to Pareto optimal vectors $\boldsymbol{x} \in \Omega$, where $\Omega \subseteq X$, is called Pareto-optimal frontier. Finding the whole Pareto-optimal frontier is a necessary condition for selecting trade-off solutions.

Regarding the problem of cyclic planning within multi-echelon supply chain environment we deal with two multiple objectives. The first one is to minimize the average total cost represented by sum of inventory holding, production and ordering costs in accordance with the following equation:

$$\text{Minimize } E[TC] = \sum_{t=1}^{T} (\sum_{j=1}^{J} CP_{j,t} + \sum_{i=1}^{I} CRM_{i,t} + \sum_{i=1}^{I} CH_{i,t}), \tag{2}$$

where $TC$ denotes the total cost, $CP_{i,t}$ denotes production cost in process $j$ per period $t$, $CRM_{i,t}$ is ordering cost at stock point $i$ per period $t$, and $CH_{i,t}$ - inventory holding cost at stock point $i$ per period $t$; $I$ and $J$ correspond to the number of stock points and processes, and $T$ defines the number of periods in the planning horizon.

The second objective is to satisfy customer service requirements specified by the order fill rate.

$$\text{Maximize } E[FR] = \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{k=1}^{K} QC_{i,k,t} \left/ \sum_{t=1}^{T} \sum_{i=1}^{I} \sum_{k=1}^{K} D_{k,i,t} \right., \tag{3}$$

where $QC_{i,k,t}$ is a fraction of orders provided by stock point $i$ to end-customer $k$ in time period $t$, $D_{k,i,t}$ is actual demand of end-customer $k$ to stock point $i$ in time period $t$.

Controlled in optimisation experiments, this performance measure is introduced to avoid unconstrained minimization of the total cost.

The decision variables are replenishment cycles and order-up-to levels. Here, cycles are considered as independent variables, and order-up-to levels are re-calculated by using simplified analytical formulas described below.

The proposed approach is to use Genetic Algorithm (GA) to guide the search towards Pareto optimal frontier. Compared to other algorithms, GAs are well suited for solving such problems as they are based on biological processes, which are inherently multi-objective.

## 3. Requirements to the solution method

The main requirements imposed on the development of GA follow from analysis of multi-objective function behaviour, a search space and specific constraints related to the investigated problem.

The multi-objective function is supposed to be (a) non-linear and non-convex; (b) non-differentiable; (c) stochastic; (d) multi-modal.

Search space of decision variables is (a) multi-dimensional; (b) with different ranges for scales.

Specific constraints can be defined as (a) use of power-of-two and nested/inverted nested inventory replenishment policies; (b) existence of feasible regions in a search space; (c) robustness of solutions.

Stochastic component of multi-objective function results in variations of objective response. Variations lead to uncertainties in decision variables. Moreover, the constraining relations between variables cause infeasible regions in search space. Because of these difficulties and the presence of local minimums, an analytical form and gradients of multi-objective function cannot be a priori derived. Therefore the most preferable technique is to develop GA in combination with the simulation model of multi-echelon supply chain to solve the investigated optimisation problem.

### 4. The proposed solution method

The simulation-based approach allows the multi-objective function (a) to be an implicit function of decision variables of the system, (b) to contain complex interactions between the system entities and (c) to be stochastic in nature.

In general, different optimisation algorithms could utilize multi-echelon supply chain simulation models for estimating values of the above-described objective function. These methods refer to the so called Simulation-Optimisation approach.

In this paper, the choice in favour of GA has been dictated by the following reasons:

- GA excellently deals with the random output of simulation experiments;
- GA can be designed to search for the entire set of Pareto optimal solutions in a single run, because it is based on evolving a population of solutions in parallel;
- GA does not make assumptions about the shape and mathematical properties of the Pareto frontier.
- GA is a heuristic search algorithm, which doesn't use gradients to find the search direction;
- GA is a global search method, which can be applied to functions with multiple local optima.

The successful performance of GA is dependent on many factors: the type and rate of crossover and mutation operators, population size, the encoding approach, etc. Currently, GA practitioners pick and adjust GA parameters empirically until they achieve adequate performance for a given problem.

The proposed Multi-Objective Simulation-based Genetic Algorithm (MOSGA) is the result of analysis and a partial modifying properties of so called Non-dominated Sorting Genetic Algorithm II (NSGA-II) and NSGA-II (Deb, Agrawal et al., 2000). This analysis refers to GA elements and parameters, such as (1) fitness assignment; (2) encoding mechanism; (3) initial population and fitness estimation through simulation; (4) population sizing; (5) crossover and mutation operators; (6) selection strategy; (7) reproduction strategy.

### 5. MOSGA description

The basic idea of MOSGA is to inherit properties of NSGA-II and adapt them to multi-echelon supply chain cyclic planning and optimisation task within the simulation-based environment.

### 5.1. Fitness assignment

In extending the ideas of GA to multi-objective cases, the first task is to accomplish fitness assignment and selection in order to guide the search towards Pareto optimal set.

In MOSGA, an NSGA-II algorithm is used to estimate fitness values of chromosomes. It has the features, such as (a) a *crowded comparison operator* ($\geq_n$) and (b) a *crowding distance measure*. The first procedure guides the selection process at the various stages of the algorithm towards a uniformly spread out Pareto-optimal frontier, and the second procedure estimates the size of the largest *cuboid* enclosing the point without including any other point in the population.

The advantage of Pareto ranking is its independence of any monotonic transformation of objective function.

### 5.2. Encoding mechanism

The encoding is aimed to produce a chromosome-like structure that will subsequently be used for breeding through the action of the various genetic operators.

In the proposed MOSGA algorithm, the binary encoding is used because of benefits, which it provides to the problem representation. Note that multi-echelon cyclic planning introduces additional constraints to decision variables, such as power-of-two and nested/inverted nested policies. In a case of the binary encoding the replenishment cycles always satisfy a power-of-two policy.

The following two-step procedure is used in MOSGA to encode replenishment cycles to a 3-bit long string that is enough to define cycles search space:
1. Represent a cycle as a multiple of two, e.g. $8 = 2^3$.
2. Encode a power to a binary string, e.g. $3 => 011$.

It could be noticed that the power '7' is the maximal one produced by a string '111'.

## 5.3. Initial population and fitness estimation through simulation

Initial population is a set of initial points used to start the search process. Ideally, it should be representative enough to cover the investigated search space. MOSGA applies the following six-step procedure to produce initial solutions.

**1. Generate randomly power-of-two cycles of each chromosome.** In order to avoid any bias at the beginning of the evolutionary run, each cycle parameter $Cy \in [1,128]$ in the initial population is initialized with a uniformly distributed random real number $R \in [0.0, \ 1.0]$.

**2. Sort cycles subject to nested (or inverted-nested) policy (optional).** Cycles provided by each chromosome are ordered according to a nested (or inverted-nested) policy by using simple *QuickSort* sorting algorithm. For example, if an initial phenotype is defined as <2, 8, 4, 1>, then a phenotype with nested policy constraints is expressed by <8, 4, 2, 1>.

**3. Calculate order-up-to levels for each chromosome using approximate analytical formulas.** Order-up-to levels $S_i$ are calculated for each phenotype using the sequence of analytical formulas presented in (Merkuryeva, Napalkova, 2007).

**4. Calculate average total costs and average fill rate of each chromosome by running a supply chain simulation model.** Average total costs and average fill rate are calculated for each chromosome in the population through simulation of a supply chain. Example of linear network simulation model is given below. Number of simulation replications and a length of warm-up period are defined using a Welch-method.

5. Define fitness values of chromosomes using a Pareto-ranking approach (rank 1 assigns the best solution). To assign fitness values to chromosomes, the above described Pareto-ranking NSGA-II algorithm is used.

**6. Order chromosomes according to their fitness values in increasing sequence.** The resulting population is sorted in increasing order according to fitness values.

## 5.4. Population sizing

The population size should guarantee both adequate genetic diversity and reasonable simulation processing time. It is chosen in such a way that every solution in the search space is attainable with the crossover genetic operator. The following formula for binary-encoded chromosomes is used in the proposed algorithm:

$$N = 1.65 * 2^{(0.2*length)}$$
(4)

where length is a length of bit string.

## 5.5. Crossover and mutation operators

One of the major issues of genetic operators is the relative importance of two genetic operators: *crossover* and *mutation* since they provide the exploration and exploitation of a search space. The exploration is used to investigate new and unknown areas in a search space. The exploitation is aimed to making use of knowledge acquired by exploration to reach better positions on the search space.

In MOSGA, the uniform crossover is introduced, and it operates in the following way:
1. Select randomly one of two parents;
2. Select a current gene of a child and make it equal to the gene of selected parent;
3. Continue until all genes are defined.

Mutation is used by GAs to diversify chromosomes in order to exploit the search space as widely as possible. Since the binary encoding is used in MOSGA, the mutation consists just in flipping random bits of the genotype.

## 5.6. Selection strategy

Selection strategy decides which areas of a search space to focus efforts on. Selection depends on (a) *comparison operator*; (b) *type of selection*.

The selection procedure of MOSGA is taken from NSGA-II algorithm. As a result, every chromosome i has two attributes, such as (a) non-domination rank, $r_i$; (b) crowding distance, $d_i$.

A crowded comparison operator $\geq_n$ is defined as follows (Deb, Agrawal *et al*., 2000):

$$i \geq_n j \text{ if } (r_i < r_j) \text{ or } ((r_i = r_j) \text{ and } (d_i > d_j))$$
(5)

Thus, between two solutions with differing non-domination ranks, the point with the lower rank is preferred. If both the points belong to the same frontier, then the point, which is located in a region with smaller number of points, is preferred (Deb, Agrawal *et al*., 2000).

The type of selection is very important for efficient investigation of a search space. It is known that proportionate selections are dependent on the objective function used. In this paper a tournament selection is used, because it applies significant selection pressure while avoiding the pitfalls of fitness ordering or ranking. Moreover, the tournament selection has the smallest time complexity.

### 5.7. Reproduction strategy

A reproduction strategy is used to control ways, in which new chromosomes are to be incorporated in the population. MOSGA applies *preserving mechanism* to avoid the loss of Pareto solutions during the evolutionary process. In each generation, a new offspring population is added to a parents' population. Pareto ranks of offspring and parents' solutions are updated. First $N$ solutions are gathered for the next generation.

## 6. Illustrative example

The above example illustrates a prototype of the MOSGA algorithm applied to the generic network supply chain introduced in business case simulation in (Merkuryev, Merkuryeva, Hatem, Desmet, 2008). Main screenshots present the network simulation model (Figure 1, left), MOSGA user interface (Figure 1, right) and summary of results (Fig. 2).

It could be noticed that the preserving mechanism allows extending the initial Pareto front, which consists of a single solution, to seven solutions in the final Pareto front. The achieved results prove the algorithm's efficiency.
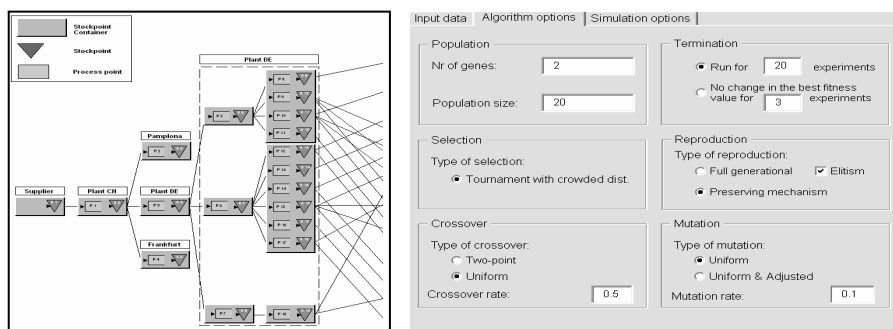


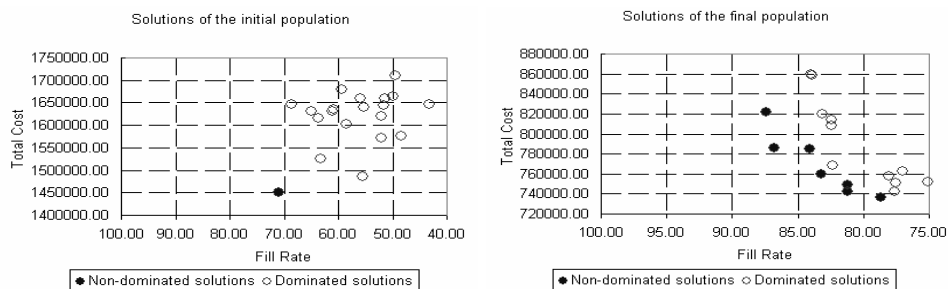**Fig. 1.** Network simulation model (left). The MOSGA user interface (right)



**Fig. 2.** Comparison between initial and final population solutions mapped in the objective space

## 3. Conclusions

The paper presents the developed multi-objective simulation-based optimisation genetic algorithm for supply chain cyclic planning and optimisation. The features of fitness assignment procedure, encoding mechanism, crossover operator, mutation operator, and selection and reproduction strategies are described in the paper. Finally, a business case numerical example that illustrates the MOSGA prototype is given.

Future research will focus on the improvement of MOSGA algorithm and its application to complex supply chain network. It is planned to include the robustness analysis in the proposed algorithm.

**References**

Chen, J.-H. 2003. Theoretical Analysis of Multi-Objective Genetic Algorithms – Convergence Time, Population Sizing and Disequilibrium, in *Report for IEEE NNS Walter Karplus Research Grant*.

Deb, K.; Agrawal, S. and Meyarivan, T. 2000. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, in *Proceedings of the Parallel Problem Solving from Nature VI Conference*.

Merkuryev, Y.; Merkuryeva, G.; Desmet, B. and Jacquet-Lagrèze, E. 2007. Integrating Analytical and Simulation Techniques in Multi-Echelon Cyclic Planning, in *Proceedings of the First Asia International Conference on Modelling and Simulation (AMS 2007)*. Phuket, Thailand.

Merkuryeva, G. and. Napalkova, L. 2007. Development of simulation-based environment for multi-echelon cyclic planning and optimization, in *6th EUROSIM Congress on Modelling and Simulation*. Ljubljana, Slovenia, 2007, paper ID 452.

Merkuryev, Y.; Merkuryeva, G.; Hatem, J. and Desmet, B. 2008. Supply Chain Simulation in the ECLIPS Project, in *Proceedings of the 2008 Asia Modelling Symposium AMS2008, IEEE*, 7  (accepted).