IRSTI 06.81.12 https://doi.org/10.26577/be-2019-4-e9

**I. Jackson**

PhD student, lecturer, Transport and Telecommunication Institute,
Latvia, Riga, e-mail: jackson.i@tsi.lv

# COMBINING SIMULATION WITH GENETIC ALGORITHM FOR SOLVING STOCHASTIC MULTI-PRODUCT INVENTORY OPTIMIZATION PROBLEM

**Abstract.** All companies are challenged to match supply and demand, and the way the company tackles this challenge has a tremendous impact on its profitability. Due to the fact that markets are rapidly evolving and becoming more complex, flexible, and information-intensive, notorious binging-and-purging approach is inappropriate. Scuh an approach, in which product is, firstly, overpurchased or overproduced in order to prepare for expected demand spikes and then discarded by sharp decline in price. Thus, in order to tailor inventory control to urgent industrial needs, the discrete-event simulation model is proposed. The model is stochastic and operates with multiple products under constrained total inventory capacity. Besides that, the model under consideration is distinguished by uncertain replenishment lags and lost-sales. The paper contains both mathematical description and algorithmic implementation. Besides that, an optimization framework based on genetic algorithm is proposed for deriving an optimal control policy. The proposed approach contributes to the field of industrial engineering by providing a simple and flexible way to compute nearly-optimal inventory control parameters.

**Key words:** stochastic inventory control, constrained optimization, simulation-optimization, genetic algorithm.

И. Джексон

докторант, оқытушы, Көлік және байланыс институты,
Латвия, Рига қ., e-mail: jackson.i@tsi.lv

## Стохастикалық мультиөнім қорларын басқару жүйесін оңтайландыру мәселелерін шешуге арналған имитациялық модельдеу мен генетикалық алгоритмнің үйлесімі

**Аңдатпа.** Әрбір компания сұраныс пен ұсынысты үндестіру қажет және компанияның осы тапсырманы қалай орындағаны оның табыстылығына үлкен әсер етеді. Нарықтың тез дамып, күрделі, икемді және өзгермелі болуына байланысты артық өндіру, сатуды ынталандыру мақсатында бағаларды күрт төмендету сияқты тәсілдер қазіргі кезде ескеріп, қолданылмайды. Осыған байланысты, бұл мақала басқарудың оңтайлы параметрлерін табу үшін қорларды басқару жүйесінің дискретті-оқиғалы имитациялық моделін жасауды көздейді. Сипатталған модель стохастикалық болып табылады және агрегаттық сыйымдылығы шектеулі бірнеше өніммен бір кезде жұмыс істейді. Сонымен қатар, қарастырылып отырған модель өнімді жеткізу арасындағы белгісіз уақыт айырмашылығымен ерекшеленеді. Мақалада математикалық сипаттама және алгоритмдік орындалу орын алады. Сонымен қатар, оңтайлы басқару саясатын табу үшін генетикалық алгоритм негізіндегі тәсіл ұсынылады. Сипатталған әдіс – қорларды басқару жүйелері үшін оңтайлы параметрлерді есептеудің қарапайым және икемді әдісі.

Ұсынылған әдіс өнеркәсіптік машина жасау саласына ықпал етеді, бірақ тәуекелдер мен сенімділік саясатына қатысты түгендеудің оңтайлы параметрлерін есептеудің қарапайым, бірақ тиімді әдісін ұсынады. Сонымен қатар, әдіс автоматтандырылған тапсырыс беру жүйесінде қолданылуы мүмкін.

**Түйін сөздер:** қорларды стохастикалық басқару, шектеулермен оңтайландыру, симуляция-оптимизация, генетикалық алгоритм.

И. Джексон

докторант, преподаватель, Институт Транспорта и Связи,
Латвия, г. Рига, e-mail: jackson.i@tsi.lv

## Комбинация имитационного моделирования и генетического алгоритма для решения задач оптимизации стохастической многопродуктовой системы управления запасами

**Аннотация.** Каждая компания сталкивается с необходимостью синхронизации спроса и предложения и то, насколько компания справляется с данной задачей, оказывает огромное влияние на ее прибыльность. В связи с тем, что рынки быстро развиваются и становятся все более сложными, гибкими, волотильными и информационно насыщенными, устаревшие подходы, предполагающие перепроизводство, сменяемое резким снижением цены для стимулирования продажи излишней продукции, являются неприемлимыми. В этой связи, данная статья преследует цель – разработать реалистичную дискретно-событийную имитационную модель системы управления запасами для нахождения оптимальных параметров контроля. Описываемая модель является стохастической и работает с несколькими продуктами при ограниченной совокупной вместимости. Кроме того, рассматриваемая модель отличается неопределенными временными лагами между поставками продукции. Статья содержит как математическое описание, так и алгоритмическую реализацию. Кроме того, для нахождения оптимальной политики управления предлагается подход на основе генетического алгоритма. Описанный метод является простым и гибким способ вычисления около-оптимальных параметров для систем управления запасами.

Предложенный подход вносит свой вклад в область промышленного проектирования, предоставляя простой, но все же эффективный способ вычисления почти оптимальных параметров запасов с учетом политики риска и надежности. Кроме того, метод может быть применен в автоматизированных системах заказа.

**Ключевые слова:** стохастическое управление запасами, оптимизация с ограничениями, симуляция-оптимизация, генетический алгоритм.

## Introduction

Modern markets are extremely competitive. Businesses are facing unceasingly growing pressure on both prices and quality. Besides that, the company is required to swiftly respond to stochastic market conditions. Incorrect inventory policy leads not only to corporate losses, but also to overproduction (Altiok, 2012). In this regard, traditional inventory policies are not appropriate anymore. Moreover, overproduction causes serious environmental problems, depleting natural resources and polluting the atmosphere.

The real-world inventory optimization is commonly characterized by the large-scale size and the necessity for nearly-optimal solutions in feasible computing times (Angun, 2011). That is why, the metaheuristics in general and genetic algorithms in particular are used so widely to define an optimal inventory policy. The world is full of uncertainty, which frequently makes classical deterministic approaches unsuitable due to excessive simplicity.

As it is mentioned in the recent research (Juan et al., 2015), real-life stochastic combinatorial optimization problems may be reformulated as a simulation in a natural way. Thus, the hybridization of metaheuristics and simulation techniques promises to be an efficient solution of stochastic inventory optimization and inventory control problems. First and foremost, the combination of simulation and metaheuristics is focused on efficiency taking into account stochastic components that may be contained either in the objective function or in the constraints. Such approaches are conventionally called simulation–based optimization or “simheuristics”. The method aims to utilize a simulation instead of an objective function in traditional form and apply the genetic algorithm to find such simulation adjustments that would lead to the optimal output. In the proposed method, the iterative searching process of the genetic algorithm has to assess the quality of individual solutions, highlighting the promising ones. Besides, real-world stochasticity may be modelled throughout the best-fit probability distribution. The distribution may be either theoretical or empirical, without the need to be approximated to normal or exponential.

Nowadays discrete-event simulation is the most dominant simulation paradigm for simulation-optimization frameworks (Gosavi, 2015). The first simulation-based optimization of inventory control system dates back to Fu and Hill (1997). The model assumes zero replenishment lead time and periodic review. The cost function comprises holding, purchasing, transportation and backlogging.

Among modern papers metaheuristic in general and genetic algorithms in particular are distinguished. For instance, researchers considered a stochastic supply chain management problem (Peirleitneret al., 2016). The problem is stated as bi-objective optimization problem. Such that overall supply chain costs are subject to minimization, while service level must be maximized. Such optimal control parameters as reorder pointsand lot sizes are derived by combining genetic algorithm with discrete-event simulation. In the same year discrete-rate simulation paradigm is used as a core to solve single-product inventory control problem (Zvirgzdina and Tolujew, 2016). In this study the model is developed in ExtendSim using inbuilt genetic algorithm to find optimal control parameters. The recent research focuses on spare part inventory control for an industrial plant. Assuming that the demand is driven by maintenance requirements, spare part provision for a single-line conveyor-like system is considered (Zahedi-Hosseini, 2018). Average cost per unit time is taken as the optimality criterion and optimization is conducted using SimRunner's inbuilt genetic algorithm.

This paper describes a possible combination of discrete-event simulation and genetic algorithm to define the optimal inventory policy in stochastic multi-product inventory systems. The discrete-event model under consideration corresponds to the just-in-time inventory control system with a floating reorder point. The system operates under stochastic demand and replenishment lead time. The utilized genetic algorithm is distinguished by a non-binary chromosome encoding, uniform crossover and two mutation operators. The proposed approach contributes to the field of industrial engineering by providing a simple, but still efficient way to compute nearly-optimal inventory parameters with regard to risk and reliability policy. Besides, the method may be applied in automated ordering systems.

## Materials and methods

First of all, the method requires designing a simulation that corresponds to the real system with a high degree of accuracy. As it is already mentioned, such a simulation will play the role of an objective function. Thus, an optimization process will be reduced to the search of the best simulation adjustments. The inventory theory at its current stage has developed a significant mathematical foundation for solving problems related to the determination of the optimal inventory policy (Zipkin, 2000). The most suitable model among considered is the model of Hopp and Spearman (2008). It is also worth noting that several distinguishing features were taken from "lost sales (r, Q) inventory control model" (Kouki et al., 2015). The considered model makes several assumptions:

– Unfulfilled demands are defined as a lost opportunity and no backlog shall be fulfilled late;

– Demand size, demand frequency and replenishment lead time are continuous random variables;

– Product of a particular type is replenished by an individual supplier.

Discrete-event simulation paradigm is chosen in order to take into account random components without a dramatic increase in system complexity at the computational level. Unlike in continuous simulation, system dynamics is not unceasingly tracked during the simulation time. Discrete-event simulation contains a list of events, such that each event takes place at a particular instant of time altering the state of the system. It is important to emphasize that there are no changes in the system between consecutive events. That is why, the simulation laps in time from previous event to the next one and runs much faster saving precious computational resources. Each event is scheduled according to preliminary generated time $t_n$ and executes sequentially. Generated time is appended to a time vector $T = (t_0, t_1 \ldots t_n)$, which may be interpreted as a time-counter. The total inventory assortment corresponds to the set of products $P$, such that each product $p_i$□$P$. The storage capacity allocation is the first priority task. Presuming that $I_{max}$ is the total storage capacity, we declare $B$ as a vector of individual storage capacities assigned for products:

$$\sum_{i=1}^{|P|} b_i = I_{max}; \ \forall \, b_i \in B \qquad (1)$$

The simulation begins with an initial inventory level of $I_p$ at $t_0$. During the simulation, emerging demands $x_{p,t}$ are are satisfied and the stock level declines gradually. If the stock level falls below a reorder point $r_{p,t}$, the inventory places a new order yp,t to refill the stock. Therefore, an inventory level at a particular moment of time equals to an inventory level in previous moment subtracting received demand and adding an order that was placed at $t - L$ Equation 2. Where Lp is the replenishment lead time for a product p. It is also worth noting that such a model aims to represent the inventory under some sort of just-in-time policy, thus, the order size $y_{p,t}$ equals to the corresponding maximal inventory capacity $b_p$ subtracting the difference between the current inventory level $I_{p,t}$ and adjusted safety-stock $SS_p$ Equation 3. In the proposed model, a new reorder point $r_p$ is recalculated after each replenishment Equation 4. Where $m_{p,[t-L,t)}$ stands for a mean demand during the replenishment lead time and $SS_p$ is a value of the corresponding safety-stock. Based on that, the number of arisen backorders for product $p$ at time $t$ may be determined as the step function:

$$I_{p,t+1} = I_{p,t} - x_{p,t} + y_{p,t-L} \qquad (2)$$

$$y_{p,t} = \begin{cases} b_p - (I_{p,t} - SS_p), if\ I_{p,t} > SS_p \\ b_p, if\ I_{p,t} \leq SS_p \end{cases} \qquad (3)$$

$$r_{p,t} = m_{p,[t-L,t)} + SS_p \qquad (4)$$

$$O_{p,t} = \begin{cases} 0, if\ x_{p,t} \leq I_{p,t} \\ x_{p,t} - I_{p,t}, if\ x_{p,t} > I_{p,t} \end{cases} \qquad (5)$$

Discrete-event simulation of such models is simple enough and can be performed by the iterative algorithm (Figure 1).

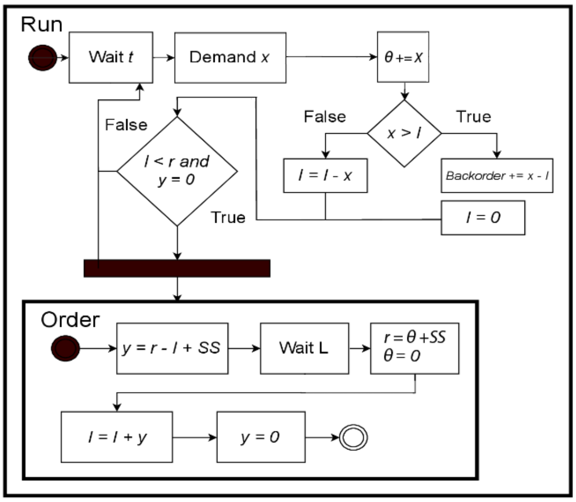

**Figure 1** – The logic behind the simulation

Each product in an assortment has a different market price and thus a different backorder cost $o_p$. Likewise, unit costs of storage and shipping, $h_p$ and $l_p$ respectively, vary depending on product's properties and subtleties of handling. Thereby, the total cost function for each product is the sum of the products of unit costs on number of units shipped, stored or backordered respectively:

$$TC_p = l_p \sum_{i=0}^{t} y_{p,t} + h_p \sum_{i=0}^{t} I_{p,t} + o_p \sum_{i=0}^{t} O_{p,t} \qquad (6)$$

In such settings, an overflow may occur:

$$F_t = \sum_{i=1}^{|P|} I_i > \sum_{i=1}^{|P|} b_i \qquad (7)$$

Such a case may be taken into account by declaring a specific cost $s$ related to the unit overflow and tracing the overflow level. In real world, such a cost corresponds to the warehouse outsourcing or reverse logistics (Bijvankand Vis I, 2011).

$$F_t = \begin{cases} 0, if\ \sum_{i=1}^{|P|} I_i \leq \sum_{i=1}^{|P|} b_i \\ F_t, if\ \sum_{i=1}^{|P|} I_i > \sum_{i=1}^{|P|} b_i \end{cases} \qquad (8)$$

In this regard the total costs function for an inventory as a whole $TC$ takes the following form

$$TC = \sum_{p=1}^{|P|} TC_p + s \sum_{i=0}^{t} F_t \qquad (9)$$

The genetic algorithm was invented and firstly introduced by Holland (1975). To date, genetic algorithms have been successfully implemented in logistics and supply chain management (Yeh and Chuang, 2011). The motivation for combining genetic algorithm with simulation is that in real-life inventory problems, it is highly preferable to obtain a nearly-optimal solution for a precisely accurate model than the absolutely optimal solution for an oversimplified deterministic model. Genetic algorithms are totally different in comparison with the conventional search techniques. The optimization procedure starts with an initial set of randomly generated solutions that are called population. Each individual solution in the population is called a chromosome. The chromosomes undergo changes through sequential iterations. Such iterations are called generations. The chromosomes within the generation are evaluated, according to a fitness function. The next generation is composed by a set of new chromosomes, called offspring. Offspring, in its turn, is mainly formed by the fittest chromosomes, partially altered by either crossover or mutation operators.

In order to apply genetic algorithm, the following initial parameters are required:

– Population size ($N$) – the number of chromosomes in each generation;

– Crossover rate ($P_c$) – the probability of executing a crossover operator;

– Mixing ratio ($P_u$) – the probability for each attribute to be exchanged;

– Mutation rate ($P_{m1}$) – the probability of executing a mutation operator 1;

– Mutation rate ($P_{m2}$) – the probability of executing a mutation operator 2;

– Mutation step (*delta*) – the gene-multiplier used by the mutation operator 2;

– Tournament size (*t*).

Practically, genetic algorithm is quite efficient in cases of large search space with lack of knowledge on the structure of the fitness function. The stochastic inventory optimization problem undoubtedly belongs to this domain. Moreover, in cases of high stochasticity, it becomes difficult to apply some traditional optimization techniques.

Genetic algorithm is quite famous as a problem-independent approach, nevertheless, the chromosome representation is a critical issue. Applying genetic algorithm to the inventory optimization problem under consideration, we are looking for such adjustments to simulation parameters: storage-resources allocation *B* and corresponding safety-stock levels *SS* that lead to the best fitness. The chromosome may be encoded as a $|P|$ size list of integers $v = (b_1, SS_1, b_2, SS_2, \dots b_{|P|}, SS_{|P|})$. In such a list each odd element stands for the inventory capacity allocated to each product p and each even element represents adjusted safety-stock level for the corresponding product *p* (Figure 2).

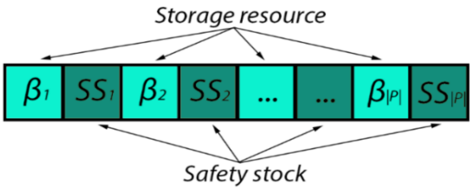

**Figure 2** –Chromosome representation

In such a simulation-driven approach, fitness function is evaluated by sequential runs of several simulations. In this case, fitness is the mean value of total costs calculated in several sequential simulation's runs. with the same parameters. We are looking for such parameters that lead to the minimal mean value of the total cost function satisfying the constraints:

$$\min_{a \in A} E[\sum_{p=1}^{|P|} TC_p(a)] \qquad (10)$$

$$\sum_{i=1}^{|P|} b_i \le I_{max}; \forall i = 1,2,3,\dots,|P| \qquad (11)$$

$$SS_i \le b_i; \forall i = 1,2,3,\dots,|P| \qquad (12)$$

In case the solution does not satisfy constraints, the fitness will take extremely high values, due to infeasibility of such a solution. During the optimization procedure, such individuals (candidate solutions) will have only an insignificant chance to pass to the next generation.

It is pointing out that a suitable chromosome representation for the particular problem domain is an extremely important task, since a good choice will make the search faster and easier by restricting the search space. However, it is tremendously important to keep in mind that the crossover and mutation operators must take into account the design of the chromosome. It is important to emphasize that in the considered problem a non-binary chromosome representation was chosen.The main reason why binary representation is the most frequent is the simplicity to implement and popularity in academic papers (Davis, 1991). Moreover, binary chromosome representation is usually space-efficient, that is why it was so popular in times, when memory was a serious problem. However, in real-world problems, it becomes common to create a genotype representation that corresponds to the considered problem with a high degree of accuracy.Crossover is the distinguishing operator of the genetic algorithm. Basically, it is a process of taking two parent solutions and production of offspring solutions in order to get a new, potentially better one. Crossover is used to vary chromosomes from one generation to the next. In order to solve the stochastic inventory problem, the uniform crossover is proposed (Figure 3).

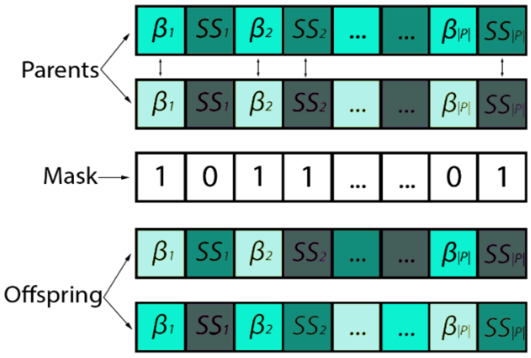

**Figure 3** –Uniform crossover representation

In the uniform crossover individual genes in the chromosome are compared between two parents and swapped with the fixed mixing ratio $P_u$. Uniform crossover is chosen for two main reasons. Firstly, since genes in the chromosome correspond to different simulation parameters *SS* and *B*, we seek a way to keep odd and even genes separated. Secondly, the uniform crossover is an efficient way to avoid the premature convergence (Michalewicz, 1996).

```
$P_u$ ← probability of swapping values
$\vec{v}$ ← first vector <$v_1$, $v_2$, ..., $v_n$>
$\vec{w}$ ← second vector <$w_1$, $w_2$, ..., $w_n$>
for$i$ in (1, $length\ of\ vector$)do
    if$P_u \geq$ $random\ number$ then
        swap the values of $v_i$and $w_i$
return $\vec{v}$ and $\vec{w}$
```

Besides, genetic algorithm requires a mutation operator to perform the optimization. Taking into account the particularities of chromosome encoding, it is proposed to apply two different mutation operators ("mild" and "radical"). The radical mutation is applied in order to prevent the premature convergence (otherwise population may get stuck in local optima). In radical mutation we replace gens in the chromosome by a new integer number in a feasible range (0, $I_{max}$) with the probability $P_{m1}$.

```
$P_{m1}$ ← probability of replacing
$\vec{v}$ ← vector
for$i$ in (1, $length\ of\ \vec{v}$) do
    if$P_m \geq$ $random\ number$then
        $v_i$← $random\ integer$
return $\vec{v}$
```

On the other hand, mild mutation is applied to accelerate convergence. The mild-mutation operator alters genes in the chromosome with the probability $P_{m2}$ by multiplying them on the step *delta* rounding to the nearest integer after that.

```
$P_{m2}$ ← probability of altering value
$\vec{v}$ ← vector
for$i$ in range (1, $length\ of\ \vec{v}$) do
    if$P_m \geq$ $random\ number$ then
        $v_i$← round($v_i$* $delta$)
return $\vec{v}$
```

It is concluded that tournament selection is an efficient and robust mechanism for working with imperfect fitness functions (Miller and Goldberg, 1995). Tournament selection runs several "tournaments" among *t* individuals (chromosomes) randomly chosen from the population. The fittest individual in each tournament is selected for the following crossover. Since weak individuals have relatively a small chance to be selected in large tournaments, it is quite important to find the optimal tournament size *t*. Tournament Selection can be programmed by the extremely simple algorithm:

```
$P$← population
$t$ ← tournament size, $t \geq 2$
$Best$← random individual from $P$
**for**$i$ in range (2 to $t$) **do**
    $Next$ ← random individual from $P$
    **if** Fit($Next$) > Fit($Best$) **then**
        $Best$← $Next$
**return** $Best$
```

Tournament selection has several significant benefits over alternative selection methods, namely, it is both simple and efficient to code, it works with parallel architectures and, lastly, it may be easily adjusted.

## Results and Discussion

Consider an example of the six-product inventory control system that operates under just-in-time policy. There is a retailer selling products of 6 types that are replenished by an individual supplier. Products of all six types share a common storage with a limited capacity of 150 pallets $\sum_{i=}^{6}$ 150. Each type of product has a unique triangular distribution for both demand size and replenishment lead time and exponential distribution for demand interarrivals.

We apply given adjustments and execute 60-days simulation of the inventory control system. As the result, the algorithm has successfully converged at the optimum in 122 generations. The optimal solution is represented by the chromosome $\vec{v}$ = (30, 4, 17, 2, 24, 4, 41, 5, 15, 1, 13, 1) with the expected total costs of $E[\sum_{p=1}^{6} TC_p(\vec{v})]$ = 2828835 USD. The fittest chromosome $\vec{v}$ stands for the following simulation adjustments: storage-resources allocation *B* = (30, 17, 24, 41, 15, 13) pallets with the corresponding safety-stock levels *SS* = (4, 2, 4, 5, 1, 1) pallets. The pivotal advantage of the involved discreet-event simulation is a possibility to perform risk and reliability analysis. Additionally, such an approach allows the researcher to plot the inventory dynamics in details and easily spot existing bottlenecks or system vulnerabilities.

Since the parameters of a genetic algorithm (especially a population size *N* and a tournament size *t*) have tremendous impact on convergence speed and a probability of premature convergence, it is very important to find a balance between search speed and premature convergence prevention.

Parameters with such a balance may be found empirically. We also can conclude that even relatively insignificant alterations in parameters of the genetic algorithm noticeably affect the convergence speed. Furthermore, some unsuccessful settings may result a premature convergence.

## Conclusion

In conclusion, the proposed optimization technique is a simple to design and computationally efficient approach to find nearly-optimal inventory policy in stochastic multi-product inventory systems. Additionally, the combination of discrete-event simulation and genetic algorithm provides a flexible method to solve complex problems with lack of knowledge on the structure of the objective function. Besides, the key advantage of such a simulation-driven approach is the possibility to trace inventory dynamics in details. It is supposed that the method may be applied in automated ordering systems by retail companies.

The research also concludes with a statement that the non-binary chromosome encoding in combination with uniform crossover and two mutation operators provide a fine balance between convergence speed and likelihood of premature convergence. There are still several minor problems to solve, such as the program-optimization of both the simulation and genetic algorithm. Moreover, it is crucially important to test the proposed approach on problems with higher dimension and compare it to alternative metaheuristic techniques. These issues are waiting to be deeply explored in a future research.

## References

Altiok T. (2012) Performance analysis of manufacturing systems. *Springer Science & Business Media*, pp. 312-320.

Angun E. (2011) A risk-averse approach to simulation optimization with multiple responses. *Simulation Modeling Practice and Theory*, vol. 19, pp. 911-923.

Bijvank M., Vis I. (2011) Lost-sales inventory theory: A review. *European Journal of Operational Research*, vol. 215(1), pp. 1-13.

Davis L. (1991) Hybridization and numerical representation. The Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold. pp. 61-71.

Fu K., Hill D. (1997) Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Transactions*, vol. 29, pp. 233-243.

Gosavi A. (2015) Simulation-based optimization. Berlin, Springer, pp. 117-129.

Holland J. (1975) Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press.

Hopp W., Spearman M. (2008) Factory Physics. Waveland Press.

Juan A., Faulin J., Grasman, S., Rabe, M., Figueria, G. (2015) A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspective*, vol. 2, pp. 62–72.

Kouki C., Jemai K., Minner S. (2015) A Lost Sales (r, Q) Inventory Control Model for Perishables with Fixed Lifetime and Lead Time. *Int. J. Production Economics*, vol. 168, pp. 143-157.

Michalewicz Z. (1996) Evolution strategies and other methods. Genetic Algorithms + Data Structures = Evolution Programs, Heidelberg, pp. 159-177.

Miller B., Goldberg D. (1995) Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, vol. 9(3), pp. 193-212.

Peirleitner K., Altendorfer A., Felberbauer T. (2016) A simulation approach for multi-stage supply chain optimization to analyze real world transportation effects. *2016 Winter Simulation Conference (WSC)*, pp. 2272-2283.

Yeh W., Chuang M. (2011) Using multi-objective genetic algorithm for partner selection in green supply chain problems. *Expert Systems with applications*, vol. 38(4), pp. 159-177.

Zahedi-Hosseini F. (2018) Modeling and simulation for the joint maintenance-inventory optimization of production systems. *2018 Winter Simulation Conference*, pp. 3264-3274.

Zipkin P. (2000) Foundations of inventory management. McGrawHill.

Zvirgzdina B., Tolujew J. (2016) Experience in Optimization of Discrete Rate Models Using ExtendSim Optimizer. *9th International Doctoral Students Workshop on Logistics*, Magdeburg, pp. 95-100.