International Journal of New Computer Architectures and their Applications (IJNCAA) 7(1): 29-36
The Society of Digital Information and Wireless Communications, 2017 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Application of Continuous Genetic Algorithms for Optimization of Logistic Networks Governed by Order-Up-To Inventory Policy

Przemysław Ignaciuk and Łukasz Wieczorek
Institute of Information Technology, Lodz University of Technology
215 Wólczańska St., 90-924 Łódź, Poland
przemyslaw.ignaciuk@p.lodz.pl, 801144@edu.p.lodz.pl

ABSTRACT

The paper addresses the optimization problem of goods distribution process in logistic networks. The controlled nodes in the considered class of networks form a mesh structure. The stock level at the nodes is replenished from external sources and other nodes in the controlled network. The external demand is imposed on any node without prior knowledge about the requested quantity. Inventory control is realized through the application of order-up-to policy implemented in a distributed way. The aim is to provide high customer satisfaction while minimizing the total holding costs. In order to determine the optimal reference stock level for the policy operation a continuous genetic algorithm is applied and adjusted for the analyzed class of application centered problems.

KEYWORDS

Logistic networks, order-up-to policy, optimization, continuous genetic algorithm, inventory management.

1 INTRODUCTION

The optimization of goods flow in logistic networks is a computationally challenging task. For this reason, the related research mainly focuses on simple network structures and topologies. The complex mathematical dependencies and delayed interaction of system components (e.g., in a practical system the goods cannot be transferred immediately among the nodes) make the numerical analysis of multi-node networks resource prohibitive. In particular, determination of the cost (or fitness) function is time consuming. Moreover, the presence of nonlinearities may lead to many local minima. In the scientific literature, the optimization of logistic systems is examined mainly in the case of fundamental configurations, e.g., when each internal node has only one goods supplier [1]. The most common types of such structures are:

- single-echelon systems [2, 3] – where a separate external provider is connected to each controlled node;
- serial interconnection [4, 5] – in which all the nodes are connected to each other in sequence;
- tree-like organization [6–8] – wherein the flow of goods proceeds along parallel paths and the external demand is placed at a focal point.

These studies are not sufficient for the current logistic systems, where the actually deployed architectures are much more complex. One may argue that, nowadays, the general availability of powerful computing machines creates new opportunities for solving realistic optimization problems. However, due to curse of dimensionality, performing extensive numerical treatment becomes possible only when an efficient method is selected, e.g., within the evolutionary computation domain [9].

The purpose of this paper is to evaluate the usefulness of genetic algorithms (GAs) in the optimization of logistic network performance when subjected to the control of the classical – order-up-to (OUT) [10] – inventory policy. The research is focused on the practical case of a system with mesh-type topology. In the analyzed structure type, a particular node – connected to multiple nodes – may play the role of supplier and goods provider to effectuate the stock replenishment decisions. The decisions are taken according to the indications of the OUT policy, deployed in a distributed way (independently at each node). The optimization objective is to determine the reference stock level (RSL) for individual nodes so that the holding costs in the entire system are minimized while at the same time a given service level is maintained.

Since the considered problem has a continuous search domain, applying the generic form of a GA would require translating the system variables (and associated operations) into the binary domain. Therefore, unlike the typical GA binary-value implementation, a continuous search space is used. Moreover, as opposed to the standard GA tuning procedures, proposed for the "artificial" optimization problems where multiple cost function evaluations are permissible [11], the long time of obtaining the fitness function value in the considered class of systems shifts the GA tuning effort towards constraining the number of iterations. The effectiveness of GA in reaching the optimal network state is evaluated in strenuous simulations.

2 SYSTEM DESCRIPTION

2.1 Actors in Logistic Processes

The paper analyzes the process of goods distribution among the nodes (warehouses, stores, etc.) of a logistic network. The nodes are connected with each other with mesh topology permitted. Each connection is characterized by two attributes:

- delivery delay time (DDT) – the time from issuing an order for goods acquisition until their delivery to the ordering node;
- supplier fraction (SF) – the percentage of ordered quantity to be retrieved from a particular source selected by the ordering node from its neighbors in the controlled network or external suppliers.

Apart from the initial stock at the nodes, the main source of goods in the network are the external suppliers. There are no isolated nodes that would not be linked to any other controlled node or external supplier, neither the nodes that would supply the stock for themselves. In addition, there is a finite path from each controlled node to at least one external source, which means that the network is connected. The system driving factor is the external demand imposed on the controlled nodes. The demand can be placed at any node and, as in the majority of practical cases [10, 12], its future value is not known precisely at the moment of issuing an order. The business objective is to ensure high customer satisfaction through fulfilling the external demand, at the same time avoiding unnecessary increase of the operational costs. Thus, the optimization purpose is to obtain a high service level at the lowest possible cost of goods storage at the nodes, i.e., minimizing the total network holding cost (*HC*).

2.2 Actor Interaction

The considered logistic network consists of N nodes n_i, where index $i \in \Theta_N = \{1, 2, \ldots, N\}$, and M external sources m_j, where $j \in \Theta_M = \{1, 2, \ldots, M\}$. The set containing all the indices $\Theta = \{1, 2, \ldots, N + M\}$. Let $l_i(t)$ denote the on-hand stock level (the quantity of goods currently stored) and $d_i(t)$ the external demand imposed on node i in period t, $t = \{0, 1, 2, \ldots, T\}$, T being the optimization time span. The connection between two nodes i and j is unidirectional, characterized by two attributes (α_{ij}, γ_{ij}), where:

- α_{ij} – the SF between nodes i and j, $\alpha_{ij} \in [0, 1]$;
- γ_{ij} – the DDT between nodes i and j, $\gamma_{ij} \in [1, \Gamma]$, where Γ denotes the maximum DDT between any two directly interconnected nodes.

Fig. 1 illustrates the operation sequence at a network node occurring in each period.

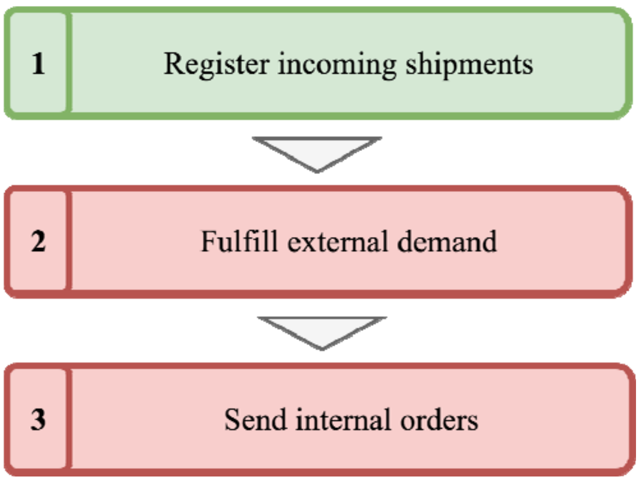

Figure 1. Node operational sequence.

Detailed mathematical description of node interaction is given in [12]. Below, only the fundamental issues required for the algorithm implementation are covered.

Let us introduce variables:

- $\Omega_i^S(t)$ – amount of goods sent by node i in period t,
- $\Omega_i^R(t)$ – amount of goods received by node i in period t.

The stock level at node i evolves according to

$$l_i(t+1) = \left(l_i(t) + \Omega_i^R(t) - d_i(t)\right)^+ - \Omega_i^S(t)\,, \quad (1)$$

where $(f)^+$ denotes the saturation function $(f)^+ = \max\{f, 0\}$. The satisfied external demand $s_i(t)$ at node i in period t (the goods actually sold to the customers) may be expressed as

$$s_i(t) = \min\left\{l_i(t) + \Omega_i^R(t), d_i(t)\right\}. \quad (2)$$

Consequently, (1) may be rewritten as

$$l_i(t+1) = l_i(t) + \Omega_i^R(t) - s_i(t) - \Omega_i^S(t)\,. \quad (3)$$

Let $o_i(t)$ denote the total quantity of goods ordered by node i in period t. $o_i(t)$ covers the orders to be fulfilled by other controlled nodes as well as external sources. Then, the quantity sent by node i in period t in response to the orders from its neighbors

$$\Omega_i^S(t) = \sum_{j\in\Theta_N} \alpha_{ij}(t) o_j(t)\,. \quad (4)$$

On the other hand, the quantity of goods received by node i in period t from all its suppliers

$$\Omega_i^R(t) = \sum_{j\in\Theta} \alpha_{ji}(t-\gamma_{ji}) o_i(t-\gamma_{ji})\,. \quad (5)$$

The nodes try to answer both the external and internal demand. In case of insufficient stock to fulfill all the requests, the ordered quantity is reduced accordingly, yet

$$\forall_i\, 0 \le \sum_{j\in\Theta} \alpha_{ji}(t) \le 1\,. \quad (6)$$

When a node receives a request from another controlled node in the network and have accumulated enough resources to fulfill it, then $\alpha_{ij}(t) = \alpha_{ij}$. Otherwise, $\alpha_{ij}(t) < \alpha_{ij}$. It is assumed that the external sources are able to satisfy every order originating from the network (they are not restricted by capacity limitation).

2.3 State-Space Description

For the purpose of convenience of further study, a state-space model of the considered network will be introduced. The dynamic dependencies can be grouped into

$$\begin{aligned} \boldsymbol{l}(t+1) = \boldsymbol{l}(t) + \sum_{\gamma=1}^{\Gamma} \boldsymbol{M}_\gamma(t-\gamma)\boldsymbol{o}(t-\gamma) \\ + \boldsymbol{M_0}(t)\boldsymbol{o}(t) - \boldsymbol{s}(t), \end{aligned} \quad (7)$$

where:

- $\boldsymbol{l}(t)$ – vector of stock levels

$$\boldsymbol{l}(t) = \left[l_1(t), l_2(t), \ldots, l_N(t)\right]^T, \quad (8)$$

- $\boldsymbol{o}(t)$ – vector of replenishment orders

$$\boldsymbol{o}(t) = \left[o_1(t), o_2(t), \ldots, o_N(t)\right]^T, \quad (9)$$

- $\boldsymbol{s}(t)$ – vector of satisfied demands

$$\boldsymbol{s}(t) = \left[s_1(t), s_2(t), \ldots, s_N(t)\right]^T, \quad (10)$$

- $\boldsymbol{M}_\gamma(t)$ – matrices specifying the node interconnections; for each $\gamma \in [1, \Gamma]$,

$$\boldsymbol{M}_\gamma(t) = \begin{bmatrix} \sum\limits_{i:\Gamma_{i1}=\gamma} \alpha_{i1}(t) & 0 & \cdots & 0 \\ 0 & \sum\limits_{i:\Gamma_{i2}=\gamma} \alpha_{i2}(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sum\limits_{i:\Gamma_{iN}=\gamma} \alpha_{iN}(t) \end{bmatrix}, (11)$$

- $\boldsymbol{M_0}(t)$ – matrix describing the stock depletion due to internal shipments

$$\boldsymbol{M}_o(t) = -\begin{bmatrix} 0 & \alpha_{12}(t) & \cdots & \alpha_{1N}(t) \\ \alpha_{21}(t) & 0 & \cdots & \alpha_{2N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N1}(t) & \alpha_{N2}(t) & \cdots & 0 \end{bmatrix}. \quad (12)$$

2.4 OUT Inventory Policy

One of the popular stock replenishment strategies applied in logistic systems is the OUT inventory policy. This policy attempts to elevate the current stock to a predefined reference one. A replenishment order is issued if the sum of the on-hand stock level and goods quantity from pending orders at a node is below the reference level. The reference level should be set so that high percentage of the external demand is satisfied, yet excessive stock accumulation is avoided. The network optimization procedures discussed in this paper provide guidelines for the reference stock level selection under uncertain demand (the future demand is not known exactly while issuing the stock replenishment orders). The operational sequence of the OUT policy is presented in Fig. 2.

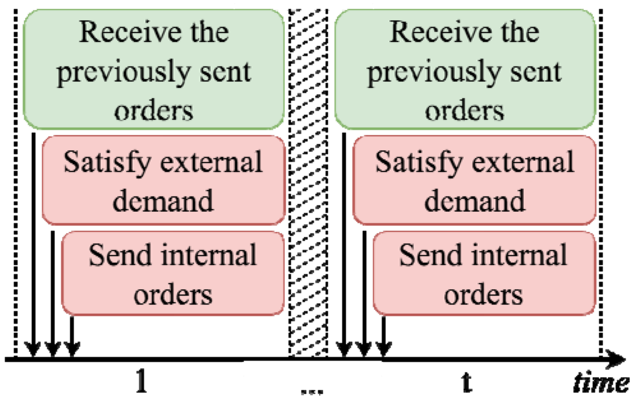

Figure 2. OUT policy operational sequence.

According to [10], the quantity of the replenishment order placed by node i in period t may be calculated as

$$o_i(t) = l_i^r - l_i(t) - \Phi_i(t), \quad (13)$$

where:

- l_i^r – the reference stock level set for node i, $i \in [1, N]$,
- $\Phi_i(t)$ – the amount of goods from pending orders issued by node i (the orders already placed by not yet realized due to lead-time delay).

In order to allow for efficient implementation of the optimization procedures variables (13) are represented in a vector form:

$$\boldsymbol{o}(t) = \boldsymbol{l}^r - \boldsymbol{l}(t) - \sum_{k=1}^{\Gamma} \sum_{\gamma=t-k}^{t-1} \boldsymbol{M}_i(\gamma)\boldsymbol{o}(\gamma), \quad (14)$$

where $\boldsymbol{l}^r$ groups the reference stock levels for all the nodes.

Any logistic system should retain a high service level despite imprecise knowledge about the demand future evolution. This objective is quantified here through the fill rate, i.e., the percentage of actually realized customer demand imposed on all the nodes. The goal of the optimization procedure is to indicate a reference stock level for each node so as to preserve the lowest possible *HC* while keeping the fill rate close to a predefined one – ideally 100%. As a first approximation, using only the knowledge about the highest expected demand in the system $\boldsymbol{d}_{\mathbf{max}}$, the 100% fill rate is obtained if the reference stock level is selected according to the following formula [12]

$$\boldsymbol{l}^r = \left(\boldsymbol{I}_N + \sum_{\gamma=1}^{\Gamma} \gamma \boldsymbol{M}_\gamma \right) \boldsymbol{M}^{-1} \boldsymbol{d}_{\max}, \quad (15)$$

where $\boldsymbol{I}_N$ is an $N \times N$ identity matrix.

3 GENETIC ALGORITHM

The key factor behind cost-efficient operation of the OUT policy is proper selection of RSL. In the analyzed class of systems it should be done simultaneously for all the controlled nodes, which is challenging due to coupled relationships in the mesh structure. GA enables automation of this process through numerical adjustments.

The continuous domain of the search space indicates direct GA application, i.e., without the typical conversion to the binary form [11]. With respect to the GA terminology, the RSL in a given node reflects the allele and the candidate solution (individual) is represented by the vector containing the RSLs of all the controlled nodes. A particular population comprises a set of such vectors. The genotypes of each individual correspond to the phenotypes of RSLs. Fig. 3

outlines the algorithm operational sequence. Its individual steps are described in latter sections.

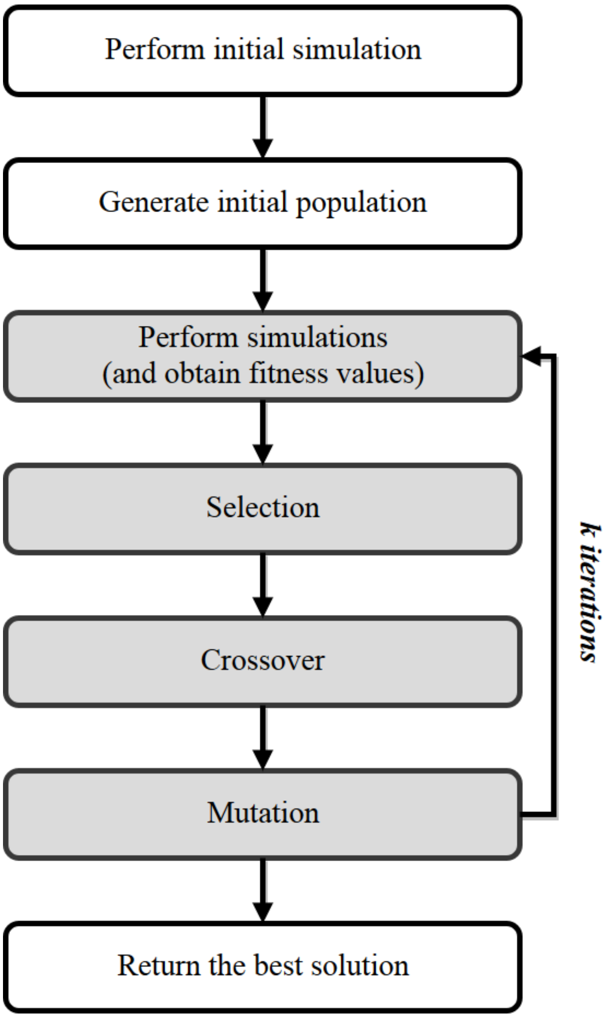

Figure 3. GA flowchart.

3.1 Initialization

The first two steps of the flowchart presented in Fig. 3 constitute the initialization phase of the optimization process. First, the RSLs are calculated according to formula (15). In the initial simulation run it is assumed that at each node a persistent external excitation (demand) is exerted during the entire simulation time equal to the maximum estimate for this node. The maximum holding cost HC_{max} and RSLs are determined accordingly and set as the boundary value for further calculations. Although the initial RSL ensures full customer satisfaction, the holding cost is very high – an excessive amount of goods is stored. The aim of further phases of the optimization process is to reduce HC while maintaining a high customer satisfaction rate.

3.2 Fitness function

A good choice of the fitness function is fundamental for GA operation and its efficiency in solving optimization problems. This function specifies dependencies between the individual system components and their relevance for the ensuing solution. It allows one to relate a particular solution to the expected optimum and influences the formation of successive populations. A well-defined fitness function should be normalized and efficient to compute. In the considered type of networks, two factors have a decisive impact on the solution:

- HC – holding cost, $HC \in [0, HC_{max}]$,
- FR – fill rate, $FR \in [0, 1]$.

HC denotes the total cost of storing goods in all the nodes throughout the simulation interval. FR provides a percentage value of how the logistic network has succeeded in fulfilling the customer demand. The goal of the optimization process is to minimize the total holding cost of the network while maintaining the highest possible fill rate. For this reason, the following fitness function has been applied to the algorithm:

$$Fitness = \left(1 - \frac{HC}{HC_{initial}}\right)^{\varphi} FR^{\beta}, \quad (16)$$

where φ and β are tuning parameters which allow one to investigate the impact of prioritizing cost reductions vs. customer satisfaction in finding the optimal solution. The fitness function enables one to measure how well adjusted is a candidate solution and compare one to another.

3.4 Selection

The selection operation in the considered GA is realized using roulette-wheel approach, also called fitness proportionate selection. It is illustrated in Fig. 4. Selection is performed once the fitness value of all the individuals in a population have been calculated. The fitness value is needed to determine the probability of choosing each chromosome in the recombination process. From the computational perspective, each individual has assigned a fraction within the range [0, 1] proportional to its fitness value relative to the rest of the current generation. Using a random selector, the entire population is divided into pairs.

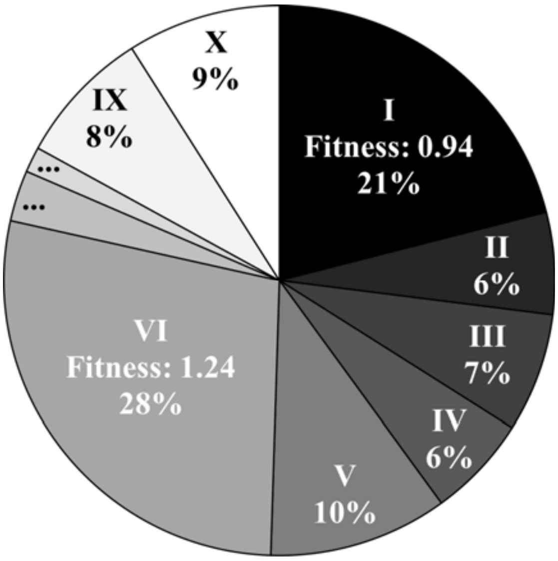

Figure 4. Roulette-wheel for a particular generation.

3.3 Crossover

The crossover operation enables evolution of populations. A pair of individuals from the source population (parents) is taken and used to obtain two child solutions forming part of the new population. For this purpose, a random natural number Δ, $\Delta \in [0, N]$, is selected. Then, each candidate solution from the parent pair is divided into two sub-vectors and two child candidate solutions are formed through swapping these sub-vectors. For two individuals $A = [l_{A1}, l_{A2}, \ldots, l_{AN}]$ and $B = [l_{B1}, l_{B2}, \ldots, l_{BN}]$ the crossover at a point Δ results in

- $C_1 = [l_{A1}, l_{A2}, \ldots, l_{A\Delta}, l_{B(\Delta+1)}, \ldots, l_{BN}]$,
- $C_2 = [l_{B1}, l_{B2}, \ldots, l_{B\Delta}, l_{A(\Delta+1)}, \ldots, l_{AN}]$.

3.4 Mutation

The final step of the GA operation is mutation. The mutation rate is defined as one of the initial parameters of GA. The mutation depends on the chosen coefficient and in the discussed implementation occurs infrequently. Basically, mutation means to replace a randomly selected gene with a random value from the considered domain. If the rate equals 0.01 each gene of the individual after crossover operation has the probability of 1% that its value will mutate.

4 NUMERICAL STUDY

In order to assess the performance of continuous GA in simulation-based optimization of logistic networks, a MATLAB-based application has been created (sources available on-line [13]). It enables one to investigate various network topologies and influence of different sets of input parameters, i.e., the number of controlled nodes and external suppliers, connectivity structure, and demand pattern. In the application, once the network structure is defined, the optimization process using either exhaustive search (for simpler topologies) or GA is performed for different RSL vectors. The obtained results are processed, logged into a text file, graphically visualized for the user.

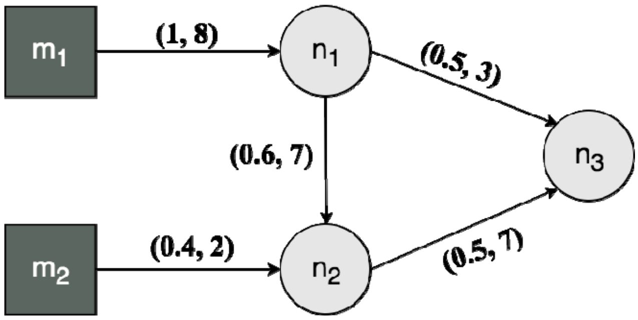

Figure 5. Logistic network scheme.

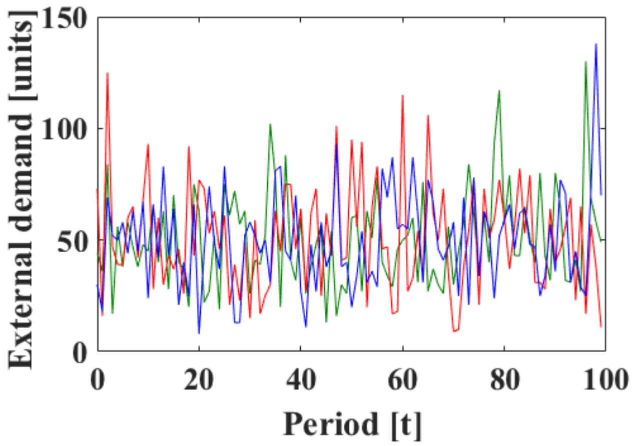

Figure 6. External demand imposed on controlled nodes.

Fig. 5 depicts the network topology selected for illustration purposes. The structure encompasses two external sources ($M = 2$) and three internal nodes ($N = 3$). The pair of numbers describing each node interconnection comprises the nominal quantity partitioning and delivery delay time (SF, DDT), e.g., node 2 acquires 60% of the established order from node 1 with lead-time delay of 7 periods. During the simulation, all the controlled nodes receive external demands. The random external demand has been generated using the Gamma distribution with parameters *shape* = 5 and *scale* = 10. The demand requests with the values in the range [8, 138] units, are illustrated in Fig. 6. The simulation lasts $T = 100$ periods. The size of the GA population has been set as 10 individuals. The overall initial holding cost, considering all three controlled nodes, equals $3 \cdot 10^5$ units.

Table 1 groups the data with regard to different fitness function shaping coefficients. In turn, the dependence between the population size and the number of iterations needed to achieve a similar optimization result is illustrated via Table 2. The presented values are the averages from multiple simulation runs taken to leverage the GA inherent randomness (sometimes the best solution has already been found in the first few iterations).

Table 1. Optimization results.

Fitness function coefficients		Optimization results		Iteration number
φ	β	Fill rate	Holding cost	
1	1	0.9890	18625	976
1	10	0.9946	30932	189
1	50	0.9987	31948	233
10	1	0.9339	10183	63
10	10	0.9833	16756	345
10	50	0.9958	25006	556
50	1	0.5445	1851	33
50	10	0.9457	10548	451
50	50	0.9881	19678	485

Table 2. Population size dependence.

Population size	Iterations needed	Time reduction
2	13184	1.0
4	4160	0.54
10	607	0.23
20	251	0.17
50	22	0.05

The analysis of the obtained data indicates that even a small change of the fitness function coefficients might have a significant impact on the holding costs, customer satisfaction, and the process of determining the optimal solution. Depending on the particular objectives, the relative importance of those factors can be modified to achieve a desirable solution. Increasing φ (equation (16)) raises the importance of holding cost reduction, whereas increasing β provides a higher fill rate (improved customer satisfaction). Simultaneous increase of both coefficients directs the system to a desirable state of maximum service rate attained with minimum holding costs. The number of iterations to reach convergence is inversely proportional to φ and grows with β.

Fig. 7 illustrates the progress of optimization process defined as the improvement of the fitness function value during successive GA iterations. The fitness function shaping parameters have been set as $\varphi = 10$ and $\beta = 50$. The graph shows that the fitness function grows fast in the first several iterations, and then improves approximately linearly. Since the optimal solution is not known *a priori*, the stopping criterion is enforced through a predefined maximum number of iterations. As the second stopping criterion, besides the simulation duration, a threshold for the number of iterations without improvement of the fitness function value is specified. In the case under consideration the threshold equals 500 iterations. The dashed line in Fig. 7 indicates the best solution established through the exhaustive search. The exhaustive search requires significantly larger number of iterations to reach the optimum than GA. It is computationally infeasible for more complex network structures.

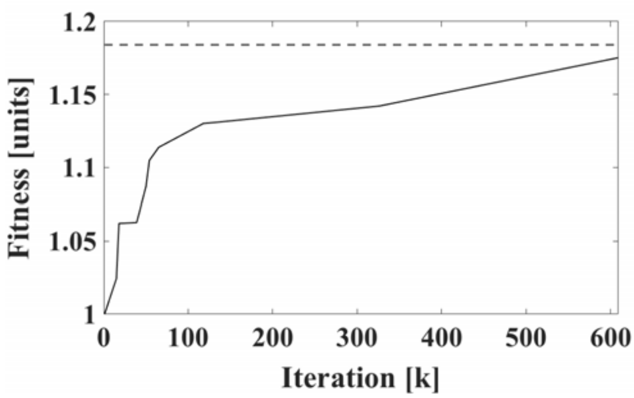

Figure 7. Fitness adjustment progress.

Figs. 8 and 9 display the stock level evolution at the controlled nodes for the initial and final (optimum) RSL setting. As can be noted from these graphs, the GA algorithm, during about

600 iterations, successfully eliminates superfluous resources. Thus, the holding cost is reduced, yet the stock level is kept positive most of the time, which implies a high fill rate.

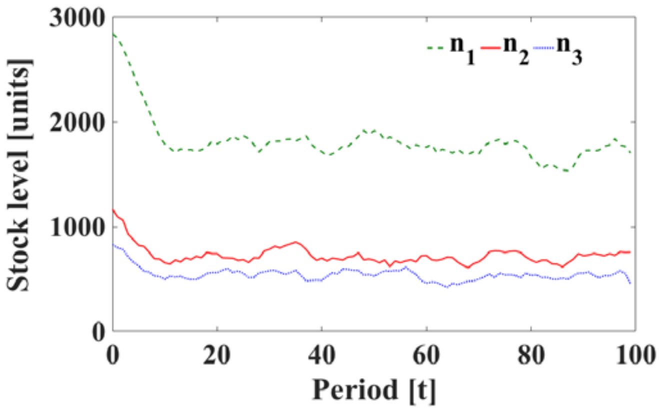

Figure 8. Stock level at the nodes for the initial generation.

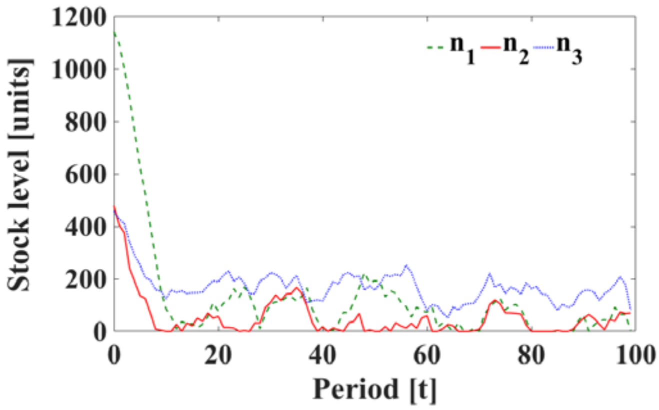

Figure 9. Stock level at the nodes for the final (optimal) generation.

4 CONCLUSIONS

The paper explores the use of continuous-domain GAs for the optimization mesh-type logistic networks governed by the OUT policy. The optimization purpose is to reduce the holding costs yet ensuring high customer satisfaction. It is achieved by adjusting the RSL at the network nodes. The fitness function of GA has been defined to allow a smooth balance between the holding costs (financial measure) and customer satisfaction through the adjustment of two algebraic coefficients. The generation size occurs to have a decisive impact on the GA operation and convergence time. The quantity of individuals in population is inversely proportional to the number of iterations needed to find the optimal solution. However, by increasing the generation size, the number of calculations and memory usage in each iteration grows fast. The numerous tests, executed for various network topologies, GA parameters, and fitness function shape coefficients, indicate that the application of GAs for RSL selection in logistic networks is advisable. As opposed to the full-search approach, the desired balance between the holding cost reduction and elevating the customer satisfaction is can be established in a reasonable time frame using common computers.

REFERENCES

1. Sarimveis, H., Patrinos, P., Tarantilis, C.D., Kiranoudis, C.T.: “Dynamic modeling and control of supply chain systems: A review,” Computers and Operations Research, vol. 35, pp. 3530--3561 (2008).
2. Hoberg, K., Bradley, J.R., Thonemann, U.W.: “Analyzing the effect of the inventory policy on order and inventory variability with linear control theory,” European Journal of Operational Research, vol. 176, pp. 1620--1642 (2007).
3. Ignaciuk, P., Bartoszewicz, A.: “Linear-quadratic optimal control of periodic-review perishable inventory systems,” IEEE Transactions on Control Systems Technology, vol. 20, pp. 1400--1407 (2012).
4. Boccadoro, M., Martinelli, F., Valigi, P.: “Supply chain management by H-infinity control,” IEEE Transactions on Automation Science & Engineering, vol. 5, pp. 703--707 (2008).
5. Movahed, K.K., Zhang, Z.-H.: “Robust design of (s, S) inventory policy parameters in supply chains with demand and lead time uncertainties,” International Journal of Systems Science, vol. 46, pp. 2258--2268 (2015).
6. Kim, C.O., Jun, J., Baek, J.K., Smith, R.L., Kim, Y.D.: “Adaptive inventory control models for supply chain management,” The International Journal of Advanced Manufacturing Technology, vol. 26, pp. 1184--1192 (2005).
7. Ignaciuk, P.: “LQ optimal and robust control of perishable inventory systems with multiple supply options,” IEEE Transactions on Automatic Control, vol. 58, pp. 2108--2113 (2013).
8. Ignaciuk, P.: “Nonlinear inventory control with discrete sliding modes in systems with uncertain delay,” IEEE Transactions on Industrial Informatics, vol. 10, pp. 559--568 (2014).
9. Chołodowicz, E., Orłowski, P.: “Comparison of a perpetual and PD inventory control system with Smith Predictor and different shipping delays using bicriterial optimization and SPEA2,” Pomiary Automatyka Robotyka, vol. 20, pp. 5--12 (2016).
10. Axsäter, S.: Inventory Control, Springer International Publishing (2015).
11. Simon, D.: Evolutionary Optimization Algorithms, John Wiley & Sons (2013).
12. Ignaciuk, P.: “State-space modeling and analysis of order-up-to goods distribution networks with variable demand and positive lead time,” Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology – ISAT 2016 – Part IV, pp. 55--65 (2017).
13. Wieczorek, Ł.: “Application sources,” [on-line]. Available: https://github.com/UksonCode/logistic-networks-ga-optimization.