Continuous GA-based Optimization of Order-Up-To Inventory Policy in Logistic Networks for Achieving High Service Rate

Przemysław Ignaciuk
Institute of Information Technology
Łódź University of Technology
Łódź, Poland
przemyslaw.ignaciuk@p.lodz.pl

Łukasz Wieczorek
Institute of Information Technology
Łódź University of Technology
Łódź, Poland
201010@edu.p.lodz.pl

Abstract—The paper addresses the problem of efficient goods distribution in logistic networks having a mesh structure. The transfer of goods takes place among the interconnected nodes with non-negligible delay. The stock gathered at the nodes is replenished from external sources as well as from other nodes in the controlled network. External demand is imposed on any node without prior knowledge about the requested quantity. The 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 at the controlled nodes a continuous genetic algorithm (GA) is applied and adjusted for the analyzed class of application centered problems.

Keywords—logistic networks, order-up-to policy, optimization, continuous genetic algorithm.

I. INTRODUCTION

The optimization of logistic network operation is a computationally challenging task. Owing to complex mathematical dependencies and delayed interaction of system components (e.g., in a practical system the goods cannot be transferred immediately among the nodes) makes 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 non-linearities may lead to many local minima. In the scientific literature, the optimization of logistic systems is examined mainly in the case of basic structures, e.g., when each internal node has only one goods supplier [1]. The most common types of such structures are:

- single-echelon [2, 3] – a single provider connected to the controlled node;
- serial interconnection [4, 5] – all the nodes connected one-by-one in a line;
- tree-like organization [6]–[8] – a particular node replenishes the stock of a few children.

These studies are not sufficient for 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 which until recently did not exist. However, 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 a sophisticated, yet realistic 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. The optimization objective is to determine the reference stock level for each individual node so that the holding costs in the entire system are minimized while at the same time maintaining a given service level.

Since the considered problem has a continuous search domain, applying basic GA would require translating the system variables (and associated operations) into the binary form. Therefore, unlike the typical GA binary-value implementation, one that resides in the 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 networked systems shifts the GA tuning effort towards constrained number of iterations. The effectiveness of GA in reaching the optimal network state is evaluated in numerous simulations.

II. SYSTEM DESCRIPTION

A. Actors in Logistic Processes

The paper analyzes the process of goods distribution among the nodes (warehouses, stores, etc.) of a logistic network. Each node has limited capacity to store the goods.
The nodes are connected in a direct manner and a mesh topology is 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 supply source selected by the ordering node.

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 customer 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 at the moment of issuing an order. The business objective is to ensure high customer satisfaction 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).

### B. 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 ($\alpha_{ij}$, $\gamma_{ij}$):

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

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

<table>
<thead>
<tr>
<th>I</th>
<th>Register incoming shipments</th>
<th>II</th>
<th>Fulfilling external demand</th>
<th>III</th>
<th>Satisfying internal orders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound</td>
<td></td>
<td>Outbound</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fig. 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:

- $\Omega^s_{ij}(t)$ – quantity of goods sent by node $i$ in period $t$,
- $\Omega^r_{ij}(t)$ – quantity of goods received by node $i$ in period $t$.

The stock level at node $i$ evolves according to

$$l_i(t+1) = l_i(t) + \Omega^s_{ij}(t) - \Omega^r_{ij}(t) - d_i(t),$$  \hspace{1cm} (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

$$s_i(t) = \min\left\{ l_i(t) + \Omega^s_{ij}(t), d_i(t) \right\}.$$  \hspace{1cm} (2)

Consequently, (1) may be rewritten as

$$l_i(t+1) = l_i(t) + \Omega^s_{ij}(t) - s_i(t) - \Omega^r_{ij}(t).$$  \hspace{1cm} (3)

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

$$\Omega^s_{ij}(t) = \sum_{j \in \Theta} \alpha_{ij}(t) o_j(t).$$  \hspace{1cm} (4)

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

$$\Omega^r_{ij}(t) = \sum_{j \in \Theta} \alpha_{ij}(t - \gamma_{ij}) o_j(t - \gamma_{ij}).$$  \hspace{1cm} (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

$$\forall \ 0 \leq \sum_{j \in \Theta} \alpha_{ij}(t) \leq 1.$$  \hspace{1cm} (6)

When a node receives a request from another controlled node in the network and is able 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 (uncapacitated external sources).

### C. State-Space Description

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

$$l(t+1) = l(t) + \sum_{j \in \Theta} M_{ij}(t - \gamma) o(t - \gamma) + M_s(t) o(t) - s(t).$$  \hspace{1cm} (7)
where the applied symbols denote:

- $I(t)$ – vector of stock levels (system state)
  \[ I(t) = [l_1(t), l_2(t), \ldots, l_N(t)]^T \]

- $o(t)$ – vector of stock replenishment orders
  \[ o(t) = [a_1(t), a_2(t), \ldots, a_N(t)]^T \]

- $s(t)$ – vector of satisfied (external) demands
  \[ s(t) = [s_1(t), s_2(t), \ldots, s_N(t)]^T \]

- $M_r(t)$ – matrices specifying the node interconnections; for each $\gamma \in [1, \Gamma]$
  \[ M_r(t) = \begin{bmatrix} 
  \sum_{i \in \gamma} a_{i1}(t) & 0 & 0 & \cdots & 0 \\
  0 & \sum_{i \in \gamma} a_{i2}(t) & 0 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & 0 & \cdots & \sum_{i \in \gamma} a_{iN}(t) 
  \end{bmatrix} \]

- $M_p(t)$ – matrix describing the stock depletion due to internal shipments
  \[ M_p(t) = \begin{bmatrix} 
  0 & a_{12}(t) & a_{13}(t) & \cdots & a_{1N}(t) \\
  a_{21}(t) & 0 & a_{23}(t) & \cdots & a_{2N}(t) \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  a_{N1}(t) & a_{N2}(t) & a_{N3}(t) & \cdots & 0 
  \end{bmatrix} \]

### D. Order-Up-To 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 level to a predefined reference value. 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. 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 precisely while issuing the stock replenishment orders). The operational sequence of the OUT policy is presented in Fig. 2.

![Fig. 2. OUT policy operational sequence.](image)

According to (10), the quantity in the replenishment order placed by node $i$ in period $t$ may be calculated as

\[ o_i(t) = I'_i - I_i(t) - \Phi_i(t), \]

where:

- $I'_i$ – the reference stock level set at node $i$, $i \in [1, N]$
- $\Phi_i(t)$ – the quantity of goods from pending orders issued by node $i$ (the orders already placed by not yet realized due to delay).

In order to allow application of the OUT policy in a distributed environment, which is considered explicitly in this work, formula (13) needs to be converted into a vector form

\[ o(t) = I' - I(t) - \sum_{\gamma=1}^{\Gamma} \sum_{k=1}^{T} M_r(\gamma) o(\gamma), \]

where $I'$ denotes the vector of reference stock levels.

A logistic network should retain a high service level despite imprecise knowledge about the demand future evolution. The system performance is quantified through the fill rate, i.e., the percentage of actually realized customer demand imposed on all the nodes. The optimization objective is to indicate a reference stock level for each node so as to preserve the lowest possible holding costs 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 $d_{\text{max}}$, the 100% fill rate is obtained if the reference stock level is selected according to the following formula

\[ I' = I_N + \sum_{\gamma=1}^{\Gamma} M_r(\gamma) M^{-1} d_{\text{max}}. \]
the domain of reference stock level selection is continuous (can take any value from a given interval), a continuous GA is employed [11]. The reference stock level at a particular node represents an allele in the population individual. As a result, one does not need to represent candidate solutions through binary sequences – the operations of selection, crossover, and mutation are executed directly for real-valued candidates.

Fig. 3 outlines the algorithm operational sequence, discussed in detail in a latter part of the paper.

Fig. 3. Genetic algorithm flowchart.

**A. Initialization**

The initialization stage includes calculating the reference stock levels according to formula (15), i.e., under the assumption that the system is faced with fixed external demand equal to its largest value throughout the entire optimization time span. This setting allows one to determine the maximum holding cost $HC_{max}$ as a boundary point for further calculations. Although full customer satisfaction is then obtained, the holding cost is high and need to be reduced.

**B. Fitness Function**

The key element in the optimization problem and evaluation of the progress of applied evolutionary method is the choice of fitness function. In the case of considered application area, two factors influence the solution fitness:

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

The simulation objective is to minimize the total system holding cost while ensuring high customer satisfaction. Accordingly, the following fitness function has been selected:

$$Fitness = \left(1 - \frac{HC}{HC_{max}}\right)^\phi FR^\beta,$$

where $\phi$ and $\beta$ are tuning parameters used to investigate the impact of prioritizing cost reductions to customer satisfaction in finding the optimal solution.

**C. Selection**

Selection in the considered GA is realized using roulette-wheel approach illustrated in Fig. 4. It means that each individual has assigned a fraction within the range $[0, 1]$ proportional to its fitness value relative to the rest of the generation. Using a random selector, the entire population is divided into pairs.

**D. Crossover**

The crossover operation is performed in the typical way for GAs. First, a uniformly distributed random number is generated. 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 determined by random number $\Delta \in [0, N]$, results in

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

Fig. 4. Roulette-wheel selection for a particular generation.
E. Mutation

The final phase of the algorithm operation is mutation, which depends on the mutation rate coefficient and occurs relatively infrequently. The mutation involves replacing a selected gene with a random value from the considered domain. If the mutation rate equals 0.01 each gene of the individual after crossover operation has the probability of 1% that its value will mutate.

IV. NUMERICAL STUDY

In order to evaluate the performance of GA in finding the optimal solution a MATLAB-based application has been created (sources available on-line [13]). The application enables generating a mesh-type network topology for given input parameters (node and supplier number, connectivity structure, and demand pattern) and obtain the fitness function value through the simulation of network behavior within a specified time frame.

Fig. 5 shows the topology under consideration in the numerical study. The network consists of two external sources ($M = 2$) and three internal nodes ($N = 3$). The numbers above the links joining the nodes indicate the nominal quantity partitioning and delay, e.g., node 3 acquires 20% of established order from node 1 with delay of 3 periods. The external demand (imposed on each node) has been generated using Gamma distribution with parameters $\text{shape} = 5$ and $\text{scale} = 10$. The simulation lasts $T = 50$ periods. The GA generation size has been set as 10. The overall initial holding cost (considering all three nodes) equals 105 units.

![Fig. 5. Logistic network structure.](image)

Table 1 groups the data regarding the fill rate and obtained holding costs for different sets of fitness function shaping coefficients. It allows one to assess the impact of cost reduction vs. ensuring high customer satisfaction.

<table>
<thead>
<tr>
<th>Fitness function coefficients</th>
<th>Optimization results</th>
<th>Iteration number</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varphi$</td>
<td>$\beta$</td>
<td>Fill rate</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.9727</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0.9994</td>
</tr>
<tr>
<td>1</td>
<td>40</td>
<td>1.0000</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.8867</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.9865</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>0.9979</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.6904</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>0.9709</td>
</tr>
<tr>
<td>10</td>
<td>40</td>
<td>0.9933</td>
</tr>
</tbody>
</table>

Fig. 6 illustrates the progress of optimization process quantified through the fitness function changes in the GA iterations. The fitness function shaping coefficients have been set as $\varphi = 10$ and $\beta = 40$. 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 improving the fitness function value is specified. In the analyzed case the threshold equals 300 iterations. The dashed line in Fig. 6 indicates the best solution established using a full search method.

![Fig. 6. Fitness adjustment progress.](image)

Figs. 7 and 8 display the stock level evolution at the nodes for the initial and final simulation. As can be seen from the graphs, the GA algorithm successfully eliminates superfluous resources (and reduces the holding costs) while keeping the stock positive most of the time, which implies a high fill rate.

![Fig. 7. Initial stock level evolution.](image)

![Fig. 8. Final stock level evolution.](image)
V. CONCLUSIONS

The paper explores the application of continuous-domain GAs for optimization of mesh-type logistic networks governed by the OUT inventory policy. In the considered class of systems, the objective has been defined as reducing the overall holding cost while ensuring high customer satisfaction by appropriate choice of the stock reference level at the network nodes. The chosen fitness function allows one to smoothly balance the economic costs and customer satisfaction by changing two algebraic coefficients. These changes have no significant effect on the number of iterations needed to find the optimal solution. The key behind the GA performance is the size of generation. By increasing the generation size, the number of iterations needed to find the optimum may be reduced. On the other hand, in each iteration the fitness function will be called more times. The tests, executed for different network structures, GA parameters and fitness function coefficients, indicate that the use of GAs for reference stock level selection is advisable. In contrast to the full-search approach, the desired balance between the holding cost reduction and elevating the customer satisfaction is achieved with computing resources attainable at a common machine.

REFERENCES


