

## A JAVA APPLICATION FOR OPTIMIZATION OF STOCK ALLOCATION IN CONTINGENCY LOGISTICS NETWORKS: COLONOR

\*\*\*

### BEKLENMEDİK DURUMLAR LOJİSTİK AĞLARINDA STOK DAĞITIMI İİN BİR JAVA UYGULAMASI: COLONOR

**Dr. Öğr. Üyesi**  
**Mehmet MİMAN**  
Harran Üniversitesi  
Mühendislik Fakültesi  
Endüstri Mühendisliği  
mmiman@harran.edu.tr  
ORCID: 0000-0002-  
1545-9960

**Hüseyin Alper TUNA**  
Toros Üniversitesi  
Mühendislik Fakültesi  
Endüstri Mühendisliği  
halpertuna@gmail.com  
ORCID: 0000-0001-  
5054-8742

**Dr. Öğr. Üyesi Mustafa**  
**DESTE**  
İnönü Üniversitesi  
İktisadi ve İdari Bilimler  
Fakültesi  
İřletme Bölümü  
mustafa.deste@inonu.edu.tr  
ORCID: 0000-0001-5781-  
6543

**Do. Dr. Gencay**  
**SARIİŐİK**  
Harran Üniversitesi  
Mühendislik Fakültesi  
Endüstri Mühendisliği  
gsariisik@harran.edu.tr  
ORCID: 0000-0002-  
1112-3933

#### Abstract

*Contingencies are unexpected events or crises that cause a major threat for security and safety of a particular population. Since they are unexpected events, the demand to perform contingency operations as well as the supply that can be provided for this can be modelled through probability distributions. Furthermore, before contingencies occur one may want to hold stocks beforehand. Based on interference theory between demand, supply and stocks, one can obtain a reliability of that site, i.e. probability that the site can perform the operations assigned based on the availability of resources for these operations. This study develops a software design as a java application, COLONOR, which optimizes the stock allocations, i.e. maximizes the reliability of contingency logistics networks with a given budget and total stocks to allocate. It assumes exponential demands and supplies, and the network structure is such that the sites are arranged either in series or parallel. The software can employ either genetic algorithm or total enumeration techniques to solve the resulting non-linear, non-separable and non-convex mathematical model and enables the users to specify the problem's parameters such as demand and supply rates, number of sites and network structure as well as the solution approach.*

**Keywords:** Contingency Logistics Networks, Stock Allocation, Genetic Algorithm, Contingency Logistics Networks Optimizer.

#### Öz

*Beklenmedik durumlar belirli bir nüfusun emniyeti ve güvenliđi için tehdit oluřturan öngörülmemiř olaylar veya krizlerdir. Bunlar öngörülemediđi için beklenmedik durumlar için gerekleřtirilecek operasyonlar için ihtiya ve o anda bunlar için sađlanabilecek tedarik olasılıđ dađımları ile modellenenebilir. Bununla birlikte beklenmedik durum oluřmadan, önceden stoklar tutulmak istenebilir. İhtiya, talep ve stok arasındaki etkileřim teorisine göre beklenmedik durumlarda görevli üssün güvenilirliđi, yani bu anlarda gerekleřtirilmesi gereken operasyonların gerekli malzemenin mevcudiyetine göre gerekleřtirilebilme olasılıđı, hesaplanabilir. Bu alıřma verilen büte ve dađıtılacak stok sayısı ile stokların dađıtımını beklenmedik durumlar lojistik ađlarının güvenirliliđini enoklayacak řekilde en iyileyen bir yazılımı java uygulaması olarak, COLONOR, geliřtirmektedir. Uygulama üssel ihtiya ve tedarik varsaymakta olup, görev üslerinin seri veya paralel řekilde konumlandırıldıđını kabul etmektedir. Yazılım neticelenen dođrusal olmayan, ayrılmayan, konveks olmayan matematiksel modeli özmek için genetik algoritma veya toplam deneme tekniklerini kullanmaktadır ve kullanıcıların talep ve tedarik oranları, operasyonel üs sayısı, ađ yapısı (seri/paralel) ve özüm tekniđi (genetik algoritma/toplam deneme) gibi problem parametrelerini belirtmesine olanak sađlamaktadır.*

**Anahtar Kelimeler:** Beklenmedik Durumlar Lojistik Ađları, Stok Dađıtımı, Genetik Algoritma, Beklenmedik Durumlar Lojistik Ađlar Eniyileycisi.

## 1. INTRODUCTION

Contingency is an “unexpected event that creates a major threat to the safety and security of a population” (Thomas, 2002; Thomas, 2004). Thomas indicates that contingencies range from military conflicts to humanitarian relief of victims from disasters such as earthquakes, hurricanes, floods, and related catastrophes. In contingency settings, operational bases are assumed to perform specific operations to recover the situations, and they require certain number of specific items to perform their duties. Thomas (2004) defines the reliability of the base from the mission perspective and once the base has enough supply, they are assumed to be reliable. As the demand required by the base as well as the amount suppliable to the base during contingencies are not known by the certainty, the reliability of the base is modeled through the interference between demand and supply, which are assumed to have certain probability distribution. Once the demand is less than supply, the base is assumed to be capable to perform the assigned mission, hence it is reliable (Thomas, 2002; Thomas, 2004; Xiong et al., 2017; Zhang et al., 2018; Tubis et al., 2017; Wang et al., 2016; Zhang et al., 2016; Zhou et al., 2016; Benyoucef et al., 2013).

Miman and Pohl (2008) extends Thomas’ work considering stocks to be hold by the base before a contingency occurs. Here, the base is assumed to be reliable if the demand it requires is less than the amount of stocks it holds plus the amount supplied to the base. They provide a stock allocation algorithm based on importance measures (Miman and Pohl, 2008; Miman and Pohl, 2012; Dağ and Miman, 2015).

Besides, they provide a set of models for contingency logistics networks (CLN) including risk distortions, practically proportional hazard and dual power distortion models developed by Offut et al. (2006) as a modeling paradigm for the risk averseness of the decision makers for a CLN design at the planning stage (Offut et al. 2006; Kang et al., 2014; Kuikka and Suojanen, 2014).

Miman (2008) provides risk displacements formulas for a variety of demand and supply distributions where the risk of the base is under the distortion. Liu et al. (2012) adapt the stocks allocation procedure proposed by Miman (2008) for spare parts inventory networks (Miman, 2008; Liu et al., 2012; Naikanet al., 2015).

There are a set of studies that considers the reliability of CLNs explicitly. Miman and Pohl (2012) provide a multi-objective optimization of CLNs through physical programming in context of selective maintenance where transportation links are assumed to have a Weibull life distribution and maintenance alternatives (do-nothing, repair, replace with an original one and replace with a superior one) effect the reliability of the links, hence the CLN. For the same selective maintenance context, Dağ and Miman (2014) provide a multi-objective optimization modeling of CLNs through utopia-distance (Miman and Pohl, 2012; Dağ and Miman, 2014; Pohl et al., 2003; Rajagopalan and Cassady, 2006; Liu et al., 2009).

Dağ and Miman (2015) in their another study proposes a multi-objective optimization modeling of the contingency logistics networks as a weighted objective function of reliability of the network, total cost of stock allocation and total number of stocks to allocate where risks in the networks in cooperated with distortions. They provide an illustrative case and its solution through the Excel Solver.

This study provides the details on a software designed to optimize stocks allocation in a CLN for the stock allocation models provided by Miman (2008) based on the genetic algorithm and total enumeration for the contingency logistics systems where the network structure is such that either bases are arranged in series or in parallel in terms of the contingency operations they perform and demand and supply are governed by exponential probability distributions. It was aimed at providing insights about contingency systems in terms of performing duties successfully and an optimizer for specified models, which can be regarded as pioneers to be extended further to deal with more complicated and realistic cases.

In presenting this study, the next section provides the mathematical model for which a software design as a JAVA application is developed which is followed by the interface of the software; details on the solution procedures, namely total enumeration and generic algorithm, and illustrative examples.

Finally, conclusion and discussion part highlight the contributions of this study and future research directions based on this study.

## 2. MODELS

### 2.1. Notation List

$S_i$	Random variable representing the supply at site $i = 1, \dots, n$
$D_i$	Random variable representing the demand at site $i = 1, \dots, n$
$X_i$	Probability distribution of the supply at site $i = 1, \dots, n$
$Y_i$	Probability distribution of the demand at site $i = 1, \dots, n$
$f_i$	Density (or probability mass) function of the supply at site $i = 1, \dots, n$
$g_i$	Density (or probability mass) function of the demand at site $i = 1, \dots, n$
$G_i$	CDF for the demand at site $i = 1, \dots, n$
$\mu_i$	Exponential rate of the supply for $Y_i$
$\lambda_i$	Exponential rate of the demand for $X_i$
$MD_i$	Mean demand at site $i$ ; $1/\lambda_i$
$MS_i$	Mean supply for site $i$ ; $1/\mu_i$
$\psi_i$	The probability distribution of excess demand for site $i = 1, \dots, n$
$s_i$	Safety stock kept at the site $i = 1, \dots, n$
$s$	Max total stock to be allocated through $n$ sites
$c_i^s$	Cost of keeping one unit stock at supply site $i = 1, \dots, n$
$\rho_i$	Failure probability of site $i = 1, \dots, n$
$R_i$	Site reliability, $i = 1, \dots, n$
$R$	Reliability of the CLN
$C$	Cost of stock allocation in CLN
$\phi^{\text{CLN}}(\bar{x}_c(k))$	structure function for the CLN
$S$	Total number of stocks to be allocated.
$N$	Total operational sites (bases) in consideration
$A$	Available number of stocks to be allocated.
$B$	Available budget for stocks allocation

### 2.2. Mathematical Formulation

According to Miman and Pohl (2008), the definition of the failure probability for the base  $i$  (probability that the base  $i$  is mission incapable due to lack of enough materials to perform the mission) can be given through Eq.1.

$$\rho_i = \Pr\{D_i - S_i \geq s_i\} = \Pr\{E_i \geq s_i\} = 1 - \psi_i(s_i) \quad (\text{Eq.1})$$

$\psi_i$  can be derived by conditioning on  $S_i$  (Assuming exponential supply and demand).

$$\begin{aligned} \psi_i(s_i) &= \Pr\{D_i < S_i + s_i\} = \int_x Y_i(x + s_i) f_i(x) = \int_0^\infty (1 - e^{-\lambda_i(s_i+x)}) \mu_i e^{-\mu_i x} dx \\ &= \int_0^\infty \mu_i e^{-\mu_i x} dx - \mu_i \int_0^\infty e^{-\lambda_i(s_i+x)} e^{-\mu_i x} dx = 1 - \mu_i e^{-\lambda_i s_i} \int_0^\infty e^{-x(\mu_i + \lambda_i)} dx = 1 - \frac{\mu_i}{\mu_i + \lambda_i} e^{-\lambda_i s_i} \end{aligned} \quad (\text{Eq.2})$$

Based on (Eq.1) and (Eq.2) the risk measure for site (base)  $i$  is given by Eq.3

$$\rho_i = 1 - \psi_i(s_i) = \frac{\mu_i}{\mu_i + \lambda_i} e^{-\lambda_i s_i} \quad (\text{Eq.3})$$

The reliability of CLN can be obtained through structure function (i.e. depending on supportability of bases for missions) That is,

$$\begin{aligned} R &= \phi^{\text{CLN}}(R_i) \\ S &= \sum_{i=1}^N s_i \\ C &= \sum_{i=1}^N s_i c_i^s \end{aligned} \quad (\text{Eq.4})$$

The optimization models for stocks allocations in a CLN can be expressed as *P1 for a series system and P2 for a parallel system*.

$$\begin{aligned} \text{Max } R &= \prod_{i=1}^N \left( 1 - \frac{\mu_i}{\mu_i + \lambda_i} e^{-\lambda_i s_i} \right) \\ \text{s.t} & \\ & \sum_{i=1}^N s_i \leq A \\ & \sum_{i=1}^N c_i^s s_i \leq B \\ & s_i \geq 0, \quad \text{int}, \forall i = 1 \dots N \end{aligned}$$

(P1: Optimization Model for Stocks Allocation in a Series System)

$$\begin{aligned} \text{Max } R &= 1 - \prod_{i=1}^N \left( \frac{\mu_i}{\mu_i + \lambda_i} e^{-\lambda_i s_i} \right) \\ \text{s.t} & \\ & \sum_{i=1}^N s_i \leq A \\ & \sum_{i=1}^N c_i^s s_i \leq B \\ & s_i \geq 0, \quad \text{int}, \forall i = 1 \dots N \end{aligned}$$

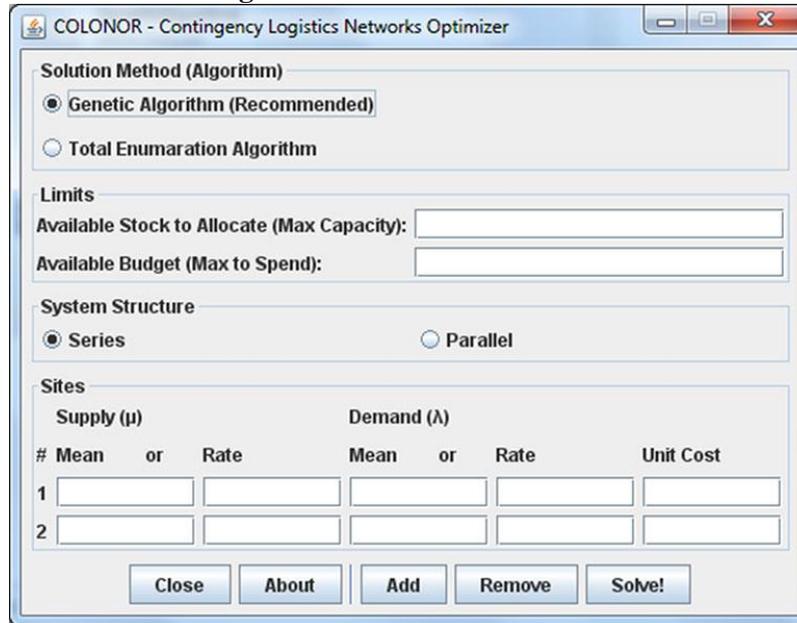
(P2: Optimization Model for Stocks Allocation in a Parallel System)

Note that both model *P1* and model *P2* are nonlinear, non-separable, non-convex in terms of decision variable  $s_i$ . Therefore, in this study, genetic algorithm is investigated as a heuristic solution to the models and the results from genetic algorithm (approximate solutions) were compared to those obtained from total enumeration (which ensures the exact optimum) for small size problems. Note that it is not possible to obtain results for a reasonable sized problem through total enumeration, hence a heuristic approach is considered to be an essential part of a contingency logistics optimizer.

### 3. INTERFACE OF THE SOFTWARE

The JAVA application developed, COLONOR v1.0 beta – Contingency Logistics Networks Optimizer beta version 1.00, in this study provides solutions to the mathematical models ( $P1$  and  $P2$ ) for stocks allocations in a CLN design and has the interface displaced in Figure 1.

Figure 1. COLONOR's Interface



The interface enables users to define the network structure either in series or parallel. Besides, the user can add/remove operational sites as much as they want by inserting related site's parameters (either mean amount or rate of demand by that site, either mean amount or rate of supply to the site, cost of holding one unit at that site). The users can specify the solution procedures (either genetic algorithm or total enumeration) whose performances to be compared in the following sections. Finally, the user can indicate the total available budget and maximum available stocks to allocate in design of the CLN. The interface provides the optimal stock allocation (amount of stocks to hold in each site) and corresponding site's and network reliability along with the how many loops the solution procedure specified uses and how much it takes in mili seconds to reach the best solution they can find. The java application developed can be obtained through the contact with authors.

### 4. AVAILABLE SOLUTION PROCEDURES

In COLONOR developed, there are two solution approaches (specifically, total enumeration and genetic algorithm) available as a solution approach to models ( $P1$  and  $P2$ ). This section provides details about each of the solution approaches along with their pseudo codes in order to enable readers to get greater insights about the java application developed.

#### 4.1. Total Enumeration Technique

Total enumeration (TE) technique aggressively tries all possibilities for decision variables  $t$  (i.e. how many stocks to hold in each operational base) through a recursive function and guarantees the optimal solution. The pseudo code of the total enumeration procedure is displayed in Exhibit-1 followed by an illustrative small-sized example about how the code works. According to Exhibit-1, initial the stock level for each site is set to zero and starting from the first site, the code increases the stock level of each site by one and compute the resulting reliabilities for each site as well as for the network along with the budget and total stocks allocated recursively. The function calls itself for the next site order (level) and so on until it reaches the last site. At the last level, it increases stock level by one and if

either budget or capacity is exceeding, it turns to the previous site (level) by returning the point it was called.

**Exhibit-1. Pseudo Code for The Total Enumeration Algorithm.**

```
Solve(0)
Function Solve (SiteOrder ):
    Set quantity 0
    Try all possibilities until reaching the limits of capacity and budget
    Do
        If parameters exceed the limits, return costs back
        If capacity or budget decreases under 0
            Increase capacity up to quantity
            Increase budget as quantity times Sites[SiteOrder][Cost]
            Set Quantities[SiteOrder] 0
            Return true
        End If
        Set Quantities[SiteOrder] as Quantity

        If the function works for last site, calculate the reliability and if it is
        higher than old one, keep this one
        If SiteOrder+1 equals Number of Sites
            Set reliability 1
            For all sites
                Calculate reliability of the current site
                Multiply the reliability with the reliability of the current
                site
            End for
            If reliability is higher than old one, or if they are similar and the
            used budget is lower than old one
                Set old one as this
                Keep the quantities as best quantities for solution
            End If
            If this is not last site, try next sites as subset
        Else If value of Solve(SiteOrder + 1) equals false
            Also return false this function
        End If

        For next loop, incerease the quantity and pay the price
        Increase quantity 1
        Decrease capacity 1
        Decrease budget as Sites[SiteOrder][Cost]
    End Do
End Function
```

The details of the total enumeration algorithm can be conceptualized better through a small example whose solution steps through TE displayed in Figure 2. The example consists of three sites arranged in series with mean supplies of 1, 2, and 2 respectively; mean demands of 2, 2, and 3

respectively with respective unit costs of stock allocation (\$4, \$3, and \$3) at these sites where there are \$10 available for the total of available stocks of 10 to allocate in the CLN.

**Figure 2.** Steps of TE based on a small-sized example

<i>Site1</i>	<i>Site2</i>	<i>Site3</i>	<i>Fitness</i>	#
0	0	1	<b>0.326</b>	1
0	0	2	0.333	2
0	0	3	0.333	3
		<i>Overflow</i>	<i>infeasible</i>	4
0	1	0	<b>0.372</b>	5
0	1	1	<b>0.609</b>	6
0	1	2	<b>0.620</b>	7
		<i>Overflow</i>	<i>infeasible</i>	8
0	2	0	0.396	9
0	2	1	<b>0.647</b>	10
		<i>Overflow</i>	<i>infeasible</i>	11
0	3	0	0.399	12
		<i>Overflow</i>	<i>infeasible</i>	13
	<i>Overflow</i>		<i>infeasible</i>	14
1	0	0	0.286	15
1	0	1	0.467	16
1	0	2	0.476	17
		<i>Overflow</i>	<i>infeasible</i>	18
1	1	0	0.534	19
1	1	1	<b>0.872</b>	20
		<i>Overflow</i>	<i>infeasible</i>	21
1	2	0	0.567	22
		<i>Overflow</i>	<i>infeasible</i>	23
	<i>Overflow</i>		<i>infeasible</i>	24
2	0	0	0.298	25
		<i>Overflow</i>	<i>infeasible</i>	26
	<i>Overflow</i>		<i>infeasible</i>	27
<i>Overflow</i>			<i>infeasible</i>	28

At the beginning of TE algorithm, the quantity of stock at each site is set to zero. Figure 2 displays how recursive loops are applied by TE. Note that there are 28 loops are called for this example where each row in Figure 2 represents a loop in the application of recursive calls by increasing the quantity of stocks by one. It starts increasing of the quantity from the last site until the site becomes overflow, i.e., this cause at least one of the constraints (in the specific example with selected parameters only budget constraint) is violated, which indicates an infeasible solution. If overflowing occurs, the next call set the quantity of the site that is overflowing to the zero and turns to the preceding site and increase the value of that site by one and recursively continues with increasing the quantity of the last site. The algorithm terminates when the first site becomes overflow (As there is no preceding site for which the quantity of stock it holds can be increased). In TE steps displayed in detail on Figure 2, the bold values indicate the best solution found so far.

#### 4.2. The Genetic Algorithm

Genetic Algorithm (GA) is an evolutionary technique that is used frequently for NP hard problems. The pseudo code for the genetic algorithm used in the java application developed is provided in Exhibit-2.

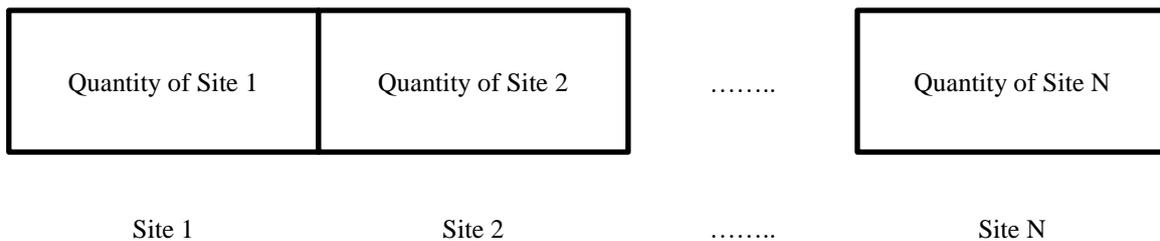
**Exhibit-2. Pseudo Code for The Genetic Algorithm**

```

Generate new random chromosomes up the population which consist of alleles that describe
parameters (Quantities of Sites)
Start Life Cycle
    Sort chromosomes of population in order to fitness function which returns the result value of
parameters owned chromosome
    If the values of chromosomes of control quantity of population is the same, break the life
cycle
    Set up new generation
    Clone the best two chromosomes from generation to new generation
    Until filling the new generation
    Do
        Select random two chromosomes in generation
        Crossing over these chromosomes from random cross-over points
        Mutating one allele of off-springs as a very low rate
        Transferring off-springs to new generation
    End Do
    Set new generation as generation
End Life Cycle
Select best chromosome from generation
Return alleles of the chromosome as parameters (Quantities of Sites)
    
```

In the GA used each solution is expressed as a chromosome which has number alleles equal to number of sites and express the quantity of stocks allocated in that site as illustrated in Figure 3.

**Figure 3.** Genotype of each chromosome



initial population is formed randomly such that the quantity of each site, expressed by each alleles, is assigned randomly according to Discrete Uniform  $\left(0, \left\lfloor \frac{B}{N} \right\rfloor\right)$  in order to start with a reasonable solution. The chromosomes in the population are arranged according to the bubble-sort. In the bubble sort, the comparisons of two adjacent chromosomes are performed according to three cases; two chromosomes express infeasible solutions, i.e. exceed at least one of the limits, one of the chromosomes expresses infeasible solution, two chromosomes express feasible solutions, i.e. satisfy all constraints. In the first case, the chromosome with a greater cost of allocation is placed into the lower position in sorting compared to the other. In the second case, the chromosome with an infeasible solution is placed to the lower position in sorting compared to the other. In the third case, the chromosome with a smaller fitness, i.e. the chromosome that has smaller over all network reliability) is placed into the lower position in sorting.

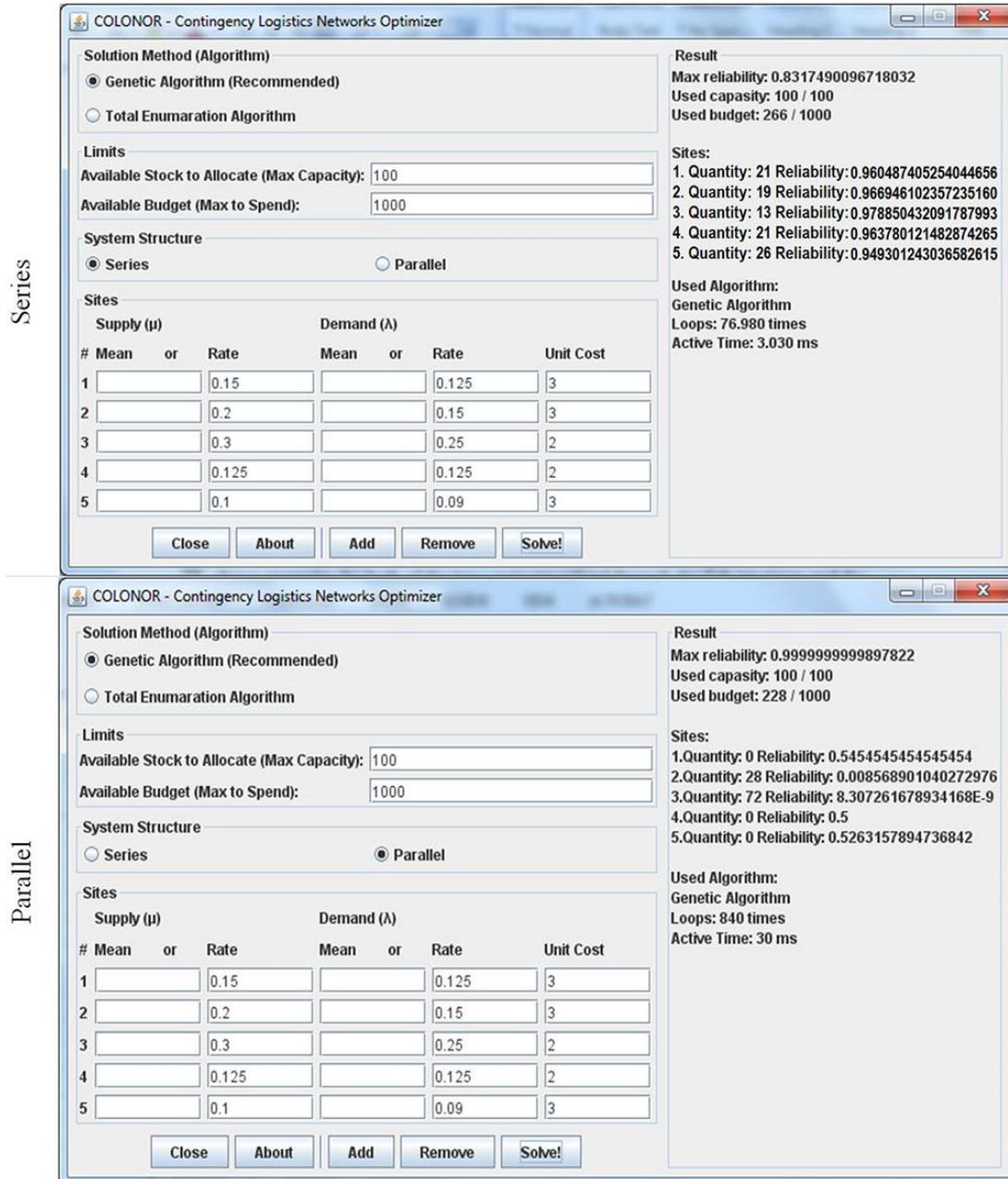
Next population in GA is generated through eliciting, crossing-over and mutation mechanisms. In eliciting the best two chromosomes (two chromosomes that have the top positions when sorted) are transferred to the next generation as they are. The rest of the next generation is obtained through crossing over and mutation mechanisms. Crossing over occurs according to single crossing over mechanism between two chromosomes that are selected randomly out of 16 chromosomes in the population where the single-crossing over point is selected randomly as well and off-springs are obtained through the exchange of alleles of parents at the crossing over point. Before off-springs are transferred to the next generation, they are subject to mutation in which one of the alleles of the off-spring is selected randomly and its value is either increased by one, decreased by one or not changed with equal probabilities, i.e 1/3. In doing this the alleles value is not allowed to fall below zero. In this case, the value, zero, is unchanged instead. The stopping criteria in the GA is convergence of the at least 14 chromosomes (control population) out of 16 chromosomes in the population to the same fitness value. Till this convergence, generation the life cycles, i.e., next generations or loops, continues. Note that the population size, and stopping criteria set here are determined based on a set of tuning efforts. In future version of the java application, these parameters can be specified by users through the interface.

## 5. ILLUSTRATIVE CASES

This section provides a small-sized illustrative CLN that consisting of five sites with demand rates 0.125, 0.15, 0.25, 0.125 and 0.09 respectively; supply rates 0.15, 0.2, 0.3, 0.125 and 0.1 respectively; cost of holding one unit at that base \$3, \$3, \$2, \$2 and \$3 respectively. There are total of 100 units available to allocate among operational sites with a total budget of \$1000. The CLN specified is considered for two cases: first the sites are arranged in series, the second the sites are arranged in parallel. Both cases are solved through the both of the available optimization techniques: total enumeration and genetic algorithm whose details provided above.

Figure 4 displays the output results obtained from COLONOR for these two cases solved through genetic algorithm to provide more conceptualization of COLONOR's interface through the results of these 2 instances.

**Figure 4.** Output Screen Snapshots for the 2 Illustrative Instances for GA



Since GA has inherent randomness, to compare the robustness and effectiveness of GA and TE, above examples for both of the two cases are solved through the GA ten times and the results (worst, best and average performances along with standard deviations) of these 10 replications are tabulated in Table 1.

**Table 1.** GA's Performances for the Illustrative Cases based on 10 Replications

Case	Performances	Worst	Best	Average	Std. Dev.
Series	Reliability	0,83174	0,83174	0,83174	0
	Number of Loops	194836	4358	63350,3	61456,2
	Time*	7680	180	2502	2418,1
Parallel	Reliability	0,99999	0,99999	0,99999	0
	Number of Loops	913	349	615,7	147,0
	Time*	40	10	24	8,4

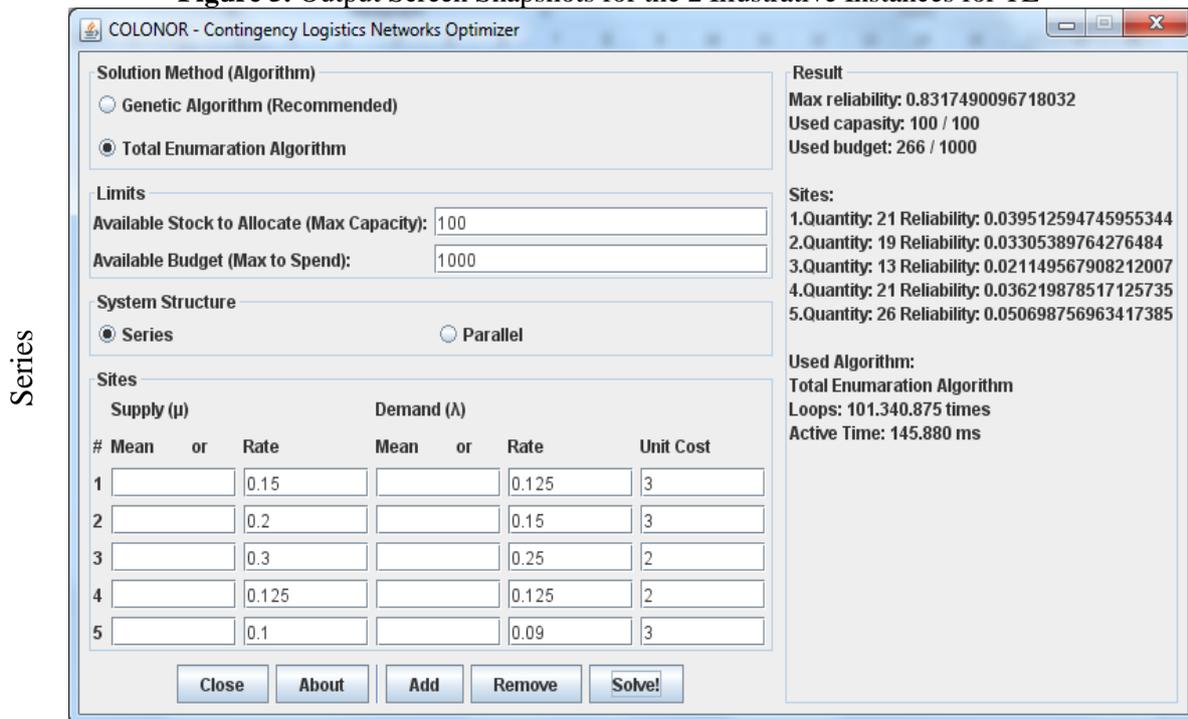
\* The values are obtained in ms through a laptop with 1.73 GHz and 4 core processor

Table 1 implies that the network structure is playing significant effect in finding the solution in terms of number of loops (generations) GA uses, hence, the time it takes to find the solution. Note that for both parallel and series systems, all the replications results in the same network reliability in 5 decimals (0,83174 and 0,99999). Therefore, the GA can be regarded as robust in terms of the objective function (network reliability). Performances of GA differs in number of generations and solution time it takes to find the evaluated best solution according to each replication. To evaluate the effectiveness of the GA, the total enumeration algorithm is used for the same two cases as well, and their exact optimal solutions are displayed in Figure 5.

According to Figure 5, the network structure (either in series or parallel) effects only the optimal stock allocation and associated network reliability in TE solution but does no effect on number of loops as total enumeration tries all possibilities exhaustively in the same way described in Figure 2. There is a slight difference in solution times which can be caused by the business of the processor at the during the algorithm run or number of operations it needs to perform in evaluating the fitness function in each of the network models described in *P1* and *P2*. That is to say that structure of the network structure in series or parallel does no effect on number of loops TE uses while negligible effect on time TE spends to find the exact optimal solution. Since there is no randomness in TE algorithm, there is no need to run replications.

The comparison of GA performances (in terms of average evaluated best solution, average number of loops used, and average time spent) with those of TE, GA provides acceptable solutions (the same stock allocations, hence, the same network reliability as those obtained TE for series network structure, different stock allocation, yet, the same network reliability in 5 decimals as those obtained in TE for parallel structure). Moreover, it is apparent that there is considerable savings GA provides in terms of number of loops it uses, and the solution time it spends to find the best evaluated solution. This is resulted from the recursive function TE uses exhaustively, hence, the number of loops TE uses as well as the solution time it needs grow exponentially as the problem size gets increased. Eventually, it is unavoidable to happen that TE cannot provide the exact solution in a reasonable time for reasonable size problems. Therefore, as seen in interfaces of the COLONOR, GA is recommended to be used for such problem as its performance are proofed to be acceptable in this study.

**Figure 5.** Output Screen Snapshots for the 2 Illustrative Instances for TE



Parallel

#	Supply ( $\mu$ )		Demand ( $\lambda$ )		Unit Cost
	Mean	or Rate	Mean	or Rate	
1		0.15		0.125	3
2		0.2		0.15	3
3		0.3		0.25	2
4		0.125		0.125	2
5		0.1		0.09	3

## 6. CONCLUSION AND DISCUSSION

This study develops a java application, COLONOR, to identify the best allocation of available stocks among the bases in the CLN under available budget to maximize the network's reliability. It assumes that each of the bases has exponential demand and supply and bases are either in series and parallel. No lateral shipment between bases is allowed.

For now there are two methods, namely total enumeration and genetic algorithm are constructed, hence available, in COLONOR to solve this non-linear, non-convex, two dimensional discrete knapsack problem. The illustrative examples, as well as the intuition indicates, imply that though TE provides exact optimum, it is cumbersome in terms of solution time and it is not possible to solve moderate and big problems using it due to its recursive nature. GA can provide approximate results that can be tolerable but may need considerable problem-specific tuning effort for its parameters.

This study can trigger a wide range of future studies. First of all, GA's parameters (number of chromosomes, stopping criteria and so on, i.e. tuning parameters for GA) can be specified by users through the interface to be developed. Further the models the COLONOR handles can be expanded including other optimization models such as the risk mitigation systems acquisition optimization model, the sustainability optimization model, joint optimization models (a set of models that are combinations of stocks allocation, risk mitigation systems acquisition, sustainability models) for the CLNs. Moreover, the assumptions in the models can be relaxed such that different probability distributions for demand and supply for a base in the network as well as any network structure depending on the truth table of base's reliability according to mission(s) perspectives can be allowed.

## CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

## REFERENCES

- BENYOUCEF, L., XIE, X. and TANONKOU, G.A. (2013). "Supply chain network design with unreliable suppliers: a Lagrangian relaxation-based approach", *International Journal of Production Research*, 51(21): 6435-6454.
- DAĞ, E. and MIMAN, M. (2014). "Beklenmedik durumlar lojistiğinin optimizasyonunda ütopya uzaklık metodu" III. Ulusal Lojistik ve Tedarik Zinciri Kongresi, Trabzon, Türkiye, 680-688.
- DAĞ, E. and MIMAN, M. (2015). "Multi-objective optimization of contingency logistics networks with distorted risks", *International Conference on Value Chain Sustainability*, İstanbul, Turkey, 487-492.
- KANG, Y., BATTÀ, R. and KWON, C. (2014). "Value-at-risk model for hazardous material transportation", *Annals of Operations Research*, 222(1): 361-387.
- KUIKKA, V. and SUOJANEN, M. (2014). "Modelling the impact of technologies and systems on military capabilities", *Journal of Battlefield Technology*, 17(2): 9-16.
- LIU, D., GAO, Q., HUANG, Z.X. and LIU, H. (2012). "Reliability of the supply and demand distribution in spare parts inventory network", In *Quality, Reliability, Risk, Maintenance, and Safety Engineering International Conference (ICQR2MSE)*, 1415-1417.
- LIU, Y., HUANG, H.Z. and ZUO, M.J. (2009). "Optimal selective maintenance for multi-state systems under imperfect maintenance", In *Reliability and Maintainability Symposium, Annual, IEEE*, 321-326.
- MIMAN, M. (2008). *Modeling and analysis of the reliability of contingency logistic networks: A multi-dimensional knapsack approach*, Phd.Thesis, University of Arkansas.
- MIMAN, M. and POHL, E. (2008). "Modelling and analysis of risk and reliability for a contingency logistics supply chain", *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 222(4): 477-494.
- MIMAN, M. and POHL, E. (2012). "Multi-objective optimisation of a contingency logistics network through physical programming", *International Journal of Collaborative Enterprise*, 3(1): 1-17.
- NAIKAN, V.N., DATAR, A. and SARMAH, S. P. (2015). "A demand-supply interference technique for modeling spare parts inventory policy", *International Journal of Management Science and Engineering Management*, 10(3): 191-198.
- OFFUT, E.J., KHAROUFEH, J.P. and Deckro, R.F. (2006) "Distorted risk measures with application to military capability shortfalls", *Military Operation Research*, 11(4): 25-39.
- POHL, E.A., CASSADY, C. R. and KWINN, M. (2003). "A selective maintenance model for serial manufacturing systems involving multiple maintenance actions", In *Proceedings of the 17th International Conference on Production Research*, Blacksburg.
- RAJAGOPALAN, R. and CASSADY, C.R. (2006). "An improved selective maintenance solution approach", *Journal of Quality in Maintenance Engineering*, 12(2): 172-185.
- THOMAS, M.U. (2002). "Supply chain reliability for contingency operations", In *Reliability and Maintainability Symposium, Proceedings, Annual, IEEE*, 61-67.
- THOMAS, M.U. (2004). "Assessing the reliability of a contingency logistics network", *Military Operations Research*, 9(1): 33-41.
- TUBIS, A., NOWAKOWSKI, T. and WERBIŃSKA-WOJCIECHOWSKA, S. (2017). "Supply chain vulnerability and resilience—case study of footwear retail distribution network", *Logistics and Transport*, 33(1): 15-24.
- WANG, Q., YUAN, X., ZUO, J., ZHANG, J., HONG, J. and LIN, C. (2016). "Optimization of ecological industrial chain design based on reliability theory—a case study", *Journal of Cleaner Production*, 124: 175-182.

- XIONG, B., LI, B., FAN, R., ZHOU, Q. and LI, W. (2017). "Modeling and simulation for effectiveness evaluation of dynamic discrete military supply chain networks", Complexity.
- ZHANG, H., WANG, M., TANG, M. and YANG, H. (2018). "The reliability measures model of multilayer urban distribution network", Soft Computing, 22(1): 107-118.
- ZHANG, H., ZHU, L. and XU, S. (2016). "Modeling the city distribution system reliability with bayesian networks to identify influence factors", Scientific Programming, 16.
- ZHOU, Q., XIONG, B., Li, B., HUANG, J. and LU, S. (2016). "Analysing the resilience of military supply network and simulation against disruptions", International Journal of Engineering Systems Modelling and Simulation, 8(3): 195-204.