Journal of Industrial and Systems Engineering
Vol. 11, Special issue: 14th International Industrial Engineering Conference
Summer (July) 2018, pp. 176-189

(IIEC 2018)
TEHRAN, IRAN

A multi-objective vibration damping meta-heuristic algorithm for multi-objective *p*-robust supply chain problem with travel time

Mahdiyar Khodemani-Yazdi[1], Reza Tavakkoli-Moghaddam[2,3,*]

[1] *Department of Industrial Engineering, Qom Branch, Islamic Azad University, Qom, Iran*
[2] *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*
[3]*Arts et Métier ParisTech, LCFC, Metz, France*

mahdiyar_khodemani@yahoo.com, tavakoli@ut.ac.ir

Abstract

The supply chain network design has a crucial role in decreasing total transportation cost. On the other hand, the value of some effective parameters, such as established facilities cost and demand, often is uncertain. In this regard, a multi-objective multi-commodity scenario-based supply chain model in the presence of disaster is proposed. Minimizing the probability of travel time exceeded at a pre-specific threshold value in different scenarios is defined as the objective function. In addition, failure probability and budget constraint can be considered as other innovations of this paper. A multi-objective vibration damping optimization (MOVDO) algorithm is developed to solve large-scale instances of the presented problem. The obtained results show that a 75-node network can be solved.

Keywords: Supply chain problem, multi-objective vibration damping optimization, travel time, budget constraint, failure rate

1- Introduction

A supply chain network consists of different parts such as suppliers, plants, warehouses and distribution centres. Also, setup cost, transportation cost, operation cost can be considered as network costs. Maximizing productivity and responsiveness has attracted more attention as two different strategies in recent years. The total network cost is tried to minimize in productivity strategy while designing a robust transportation network is considered in responsiveness strategy. In this regard, we proposed a scenario based supply chain model in presence of disaster to minimize total network cost as well as maximizing network robustness. The main contribution of this research can be mentioned below:

- Proposing a *p*-model scenario based supply chain problem.
- Developing the MOVDO algorithm to solve the model.
- Considering chance constraint in the objective function as robust measure.
- Considering budget constraint in the proposed model.

The rest of this research is organized as follows. Section 2 consists of a review of the literature on uncertain supply chain models. A multi-commodity scenario based supply chain model in the

---

*Corresponding author

ISSN: 1735-8272,

presence of disaster is formulated in section 3. In section 4, the MOVDO algorithm is presented as one of our main contribution. In section 5, computational results are demonstrated. Finally, conclusions and suggestions for future research are discussed in the last section.

2- Literature review

In this section, the previous studies on the related research are briefly surveyed. Amiri (2006) proposed a supply chain model to minimize the setup cost, in which all demands were covered. The facilities location, capacities and transportation links can be mentioned as obtained results. Pishvaee et al. (2010) presented a close-loop supply chain problem in form of a mixed-integer programming (MIP) model. A state-of-the-art paper was constructed by Melo et al. (2009). Thanh et al. (2008) surveyed a multi-commodities capacitated supply chain model and proposed defective products assumption and capacity constraint as their research innovations. Hinujusaet et al. (2008) proposed a multi-commodity supply chain problem, in which a plant can be closed during periods. Kanan et al. (2010) designed a close-loop supply chain network consisting of different levels, such as suppliers, plants, and distribution centers, retailers, scrapping centers and dismantling centers. Pishvaee et al. (2011) used a robust approach to model a supply chain network problem in form of an MIP model. Melo et al. (2003) proposed a multi-periods and multi-commodities supply chain problem, in which demand leakage can be covered using plants. Anaraki et al. (2011) considered lead-time in supply chain problem and located plants and warehouses. Daski and Verter (2001) considered supply chain network in a continuous area so that each facility serves a specific demand area. Optimizing the number of plants and number of areas were considered to cover all demand areas. Mastrocinque et al. (2013) modelled a supply chain problem to minimize the total network cost and lead-time. An artificial bee colony algorithm was applied to solve the model.

Nozick (2001) introduced an MIP model for the supply chain network design to minimize the total transportation cost. Syarif et al. (2002) surveyed the supply chain network design consisting of plants and distribution centers. Zhou et al. (2002) proposed a supply chain problem to minimize the maximum travel time, in which a trade-off between transportation cost and service level is considered. Mokashi et al. (2003) considered a supply chain problem with the aspect of distribution planning. A decomposition approach was employed to solve the model. Rabbani and Yousefnejad (2013) applied a graph theory-based approach to a supply chain problem. Pan and Nagi (2013) investigated a multi-period supply chain model considering an agile manufacturing scenario. Considering the previous studies, it appears that most of the researchers focused on minimizing the total cost in deterministic models; however, real situations are different. In this regard, we propose a supply chain model in the presence of disaster.

3- Problem statement

Due to the lack of information about the effective parameters in real-word applications, an uncertain form of the model is proposed to design a robust network. Moreover, time plays a critical role in disaster, thus a supply chain problem is modeled to minimize the probability that the travel times are not less than a given threshold value. The network cost is limited by a budget constraint. Because of considering risk, a failure matrix related to the capacity constraint is defined in the model as the contribution of this paper. It expects that the robust supply chain network design can be achieved. The proposed model is described below:

Indices	
I	Potential warehouse nodes
J	Potential distribution centre nodes
K	Relief commodities
S	Scenarios

Parameters	
p_s	Probability of the occurrence of scenario s ($s \in S$)
t_{ij}^{ks}	Stochastic travel time from warehouse i to distribution center j for commodity k under scenario s.
F_i^s	Stochastic warehouse stablished cost in node i under scenario s
ρ_i^{ks}	Stochastic ratio of inventory for warehouse i related to relief commodity k under scenario s.
τ_j^{ks}	Stochastic penalty cost of an unmet relief commodity k at distribution center j under scenario s.
γ_i^{ks}	Stochastic capacity of warehouse i related to relief commodity k under scenario s.
h^k	Unit maintenance cost for relief commodity k.
d_j^{ks}	Stochastic demand of relief commodity k in distribution center j under scenario s.
ξ^k	Maximum value of relief commodity k.
Ψ	Maximum variation for demand shortage
Cap	Capacity
Δ_{pq}	Maximum acceptable dispersion for customer satisfaction level between two demand points p and q.
T_{ij}^s	maximum allowed time corresponding to path i to j.
δ_s	Maximum allowed budget

Decision variables	
x_{ij}^{ks}	Stochastic flow corresponding to relief commodity k from warehouse I to distribution center j under scenario s.
q_i^{ks}	Stochastic pre-stored relief commodity k at warehouse i under scenario s.
z_i^{ks}	Stochastic unused relief commodity k at warehouse i under scenario s.
w_j^{ks}	Stochastic shortage of relief commodity k at distribution centre j under scenario s.
φ_j	Ratio of shortage of relief commodities at distribution centre j.
n_i^s	Number of vehicles at warehouse i under scenario s.
y_i^s	1 if a warehouse facility is located on nod i under scenario s; 0, otherwise.

The proposed problem is now formulated in a form of the scenario-based MIP model.

$$\underset{s \in S}{\text{Min}}\ p\left(\sum_{j=1}^{J}\sum_{i=1}^{I}\sum_{k=1}^{K} X_{ij}^{ks}\ \ t_{ij}^{ks} \geq T_{ij}^{s}\right) \tag{1}$$

s.t.

$$\sum_{i=1}^{I} F_i^s\ y_i^s\ +\sum_{i=1}^{I}\sum_{k=1}^{K} h^k z_i^{ks} +\sum_{j=1}^{J}\sum_{k=1}^{K}\tau_j^{ks}\ W_j^{ks} \leq \sigma_s \quad ;\ \forall s \tag{2}$$

$$d_j^{ks} = W_j^{ks} + \sum_{i=1}^{I} X_{ij}^{ks}\ ;\ \forall j \in J,\ k \in K,\ s \in S \tag{3}$$

$$q_i^{ks} \leq y_i^s\ \ \gamma_i^{ks} \quad \forall i \in I,\ \ k \in K,\ s \in S \tag{4}$$

$$\sum_{s=1}^{S}\ \sum_{i=1}^{I}\ p^s q_i^{ks} \leq \xi^k \quad \forall k \in K \tag{5}$$

$$\sum_{j=1}^{J} x_{ij}^{ks} + Z_i^{ks} = \rho_i^{ks} q_i^{ks} \quad ; \forall i \in I,\ \forall j \in J,\ k \in K,\ s \in S \tag{6}$$

$$-\Delta pq \le \Phi_p - \Phi_q \le \Delta pq \quad \forall p,q \in \{1...j\} \quad p \ne q \tag{7}$$

$$\Phi_j = \sum_{s}^{S} \sum_{k}^{K} \tau_j^{ks} w_j^{ks} \quad \forall j \in J \tag{8}$$

$$\sum_{k}^{K} w_j^{ks} \le \Psi \sum_{k}^{K} d_j^{ks} \quad \forall j \in J, s \in S \tag{9}$$

$$\frac{1}{Cap} \sum_{j}^{J} \sum_{k}^{K} x_{ij}^{ks} = n_i^{s} \quad \forall i \in I, s \in S \tag{10}$$

$$x_{ij}^{ks}, q_i^{ks}, z_i^{ks}, w_j^{ks}, n_i^{s} \ge 0 \quad \forall\ i,j,k,s \tag{11}$$

$$y_i^{s} \in \{0,1\} \quad \forall\ i,s \tag{12}$$

The probability that the travel time is not less than a given threshold value is minimized in the objective function as one of our main contributions in equation (1). Constraint (2) defines budget constraint in which it guarantees that the total network cost including setup and transportation costs does not exceed a given threshold value. Constraint (3) limits the amount of the lost demand and the products shipped to distribution centers. Constraint (4) is a capacity constraint that restricts warehouse capacity using a failure matrix. Constraint (5) control the amount of available storage relief commodities. Constraint (6) guarantees a balance between transported and stored commodities. Constraint (7) restricts the level of customer satisfaction. The level of customer satisfaction is measured by the amount of the shortage commodities as constraint (8). Maximum unmet commodities are restricted by constraint (9). The number of vehicles used in the network is calculated according to the capacity of the warehouse and the amount of flowed commodities that is guaranteed by constraint (10). Finally, constraints (11) and (12) define the decision variables.

A non-linear structure of the proposed model motivates us to make it linear for higher solvability. On the other hand, the following normal distribution can be assumed for the parameters if the number of scenarios is large enough.

$$\sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{k=1}^{K} X_{ij}^{ks} \; t_{ij}^{ks} \sim N\left(\sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{k=1}^{K} X_{ij}^{ks} \; \bar{t}_{ij}^{ks} \;,\; \sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{k=1}^{K} X_{ij}^{ks} \; \sigma_{ij}^{ks\,2} \right)$$

Now, Equation (1) can be reformulated by:

$$\min_{s \in S} \frac{T_{ij}^{s} - \sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{k=1}^{K} X_{ij}^{ks} \; \bar{t}_{ij}^{ks}}{\sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{k=1}^{K} X_{ij}^{ks} \; \sigma_{ij}^{ks\,2}} \tag{13}$$

4- Numerical results

One solution method for optimization problems is to consider all the possible solutions, calculating the objective function for all of them, and finally choosing the best solution. Obviously, a global enumeration method will lead to find the optimal solution for the given problem. However, in practice, it is impossible to apply such a method because there are too many feasible solutions. Due to the difficulties of the global enumeration method, creating more efficient and more effective method is emphasized and noticed. Various algorithms have been created including the famous simplex method in solving linear programming problems, and branch-and-bound for integer programming. Recently, there has been more concentration on heuristic, meta-heuristic, or random search methods. Meta-heuristic methods can present a satisfactory (near-optimal) solution for a problem in a limited time. They are mainly based on enumeration methods; however, they use extra information for directing the search. Such information is general in an application field and can solve very complex problems. Most of these methods are random and inspired from the nature.

4-1- Multi-objective vibration damping optimization

The vibration damping optimization (VDO) algorithm is a meta-heuristic algorithm which is developed using vibration damping concepts in vibration theory. In this research, multi-objective mode of VDO is developed for the proposed problem in discrete space. Chromosome structure proposed by researchers is used for coding. Using chromosome structure we can easily prohibit infeasible and illegitimate solutions. In other words, most of the model constraints are met by using this proposed structure and others are resolved by penalty function. The concept of VDO was first crafted and presented by Mehdizadeh and Tavakkoli-Moghaddam (2009) which is derived from the process of mechanical vibrations damping.

In order to solve the proposed model, we use the multi-objective version of the algorithm called MOVDO, which is first presented by Hajipour et al. (2014). For this purpose, two concepts of multi-objective meta-heuristic algorithm, called fast non-dominated sorting (FNDS) and crowding distance (CD), are executed which is show in figure 1.

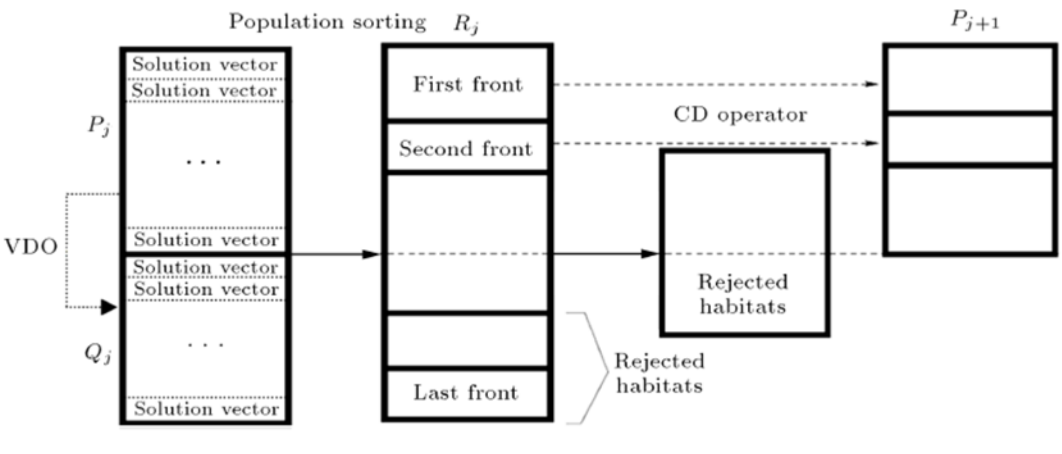

Fig 1. Evolution process in the MOVDO algorithm

In the FNDS, the R value for the initial population is compared and sorted. To do this, all chromosomes from first non-dominated layer are found. Therefore, both objective functions of the mathematical model are minimized. Chromosomes are selected based on a dominance concept. Then, in order to find the next layer of non-dominated chromosomes solutions selected for the previous layer are temporarily ignored. This is iterated until all chromosomes are put into layers and f_k (k =1, 2, …, M) will be the number of non-dominated solutions in a given layer after sorting the population. The CD value is determined to evaluate layer solutions based on a relative density of each solution. Calculation steps for the CD are stated. In order to select solutions of next generation, a tournament is executed according to the steps stated below:

- Randomly select n solutions from the population.

- Calculate the non-dominated rank of each solution and specify CD value for solutions with equal ranks.
- Select solutions with the lowest ranks. In case of multiple solutions with equal rank, select the solution with a higher CD.

In other words, the criterion of comparison for MOVDO solutions can be considered below:
If $r_y < r_x$ or ($r_y = r_x$ and $d_y < d_x$) then $y < x$, where r is the rank and d is the CD value

Considering these concepts and operators, the population of parents and children must be combined to assure Elitism. Since the combined population is bigger than the initial one, sorting non-dominated solutions must be performed again. In fact, chromosomes with higher ranks are selected and added to the population until population size reaches N. The algorithm is iterated until the determined number is obtained (or any other stopping condition occurs).

The process is started by initializing the population from solution vector p_j. Then, new operators are performed on p_j to produce new population Q_j. For elitism p_j and Q_j are combined in algorithm. In this step, R_j vectors are sorted in several layers based on the FNDS and the CD. By using the proposed selection method, a population is selected from the next iteration p_{j+1} to keep the population size constant.

The parameters of the MOVDO algorithm includes a number of iterations, population size, primary domain (A_0), maximum iteration number for each domain (L), damping coefficient (γ) and standard deviation. The number of iterations and population size have great effect on the quality of solutions obtained by the proposed algorithm. Lower values for these parameters decrease the solution time. However, the obtained solutions can be of low quality. On the other hand, high values for them can improve the quality of solutions; however, more time will be required. The initial domain and damping coefficient are effective on the probability of accepting worse solutions. As stated earlier, the probability of accepting the worse solution is as follows:

$$1 - \exp(-\frac{A^2}{2\sigma^2}) \tag{14}$$

where A is the domain in each iteration, which is calculated from the following equation.

$$A = A_0 \exp(-\frac{\gamma t}{2}) \tag{15}$$

It is clear that in lower iterations, the domain value is greater and it increases the probability of accepting the worse solutions. Increasing the number of iterations, domain decreases and so the probability of accepting the worse solutions. The damping coefficient parameter has a direct impact on domain value of each iteration. Its increase decreases the domain and the probability of accepting worse solutions.

Algorithm parameters must be set so that the probability of accepting worse solutions for initial iterations is high, but with further iterations this probability decreases. This enables the initial iterations to search the space widely, and in final iterations, the algorithm shows congruence to proper solutions. It is better that instead of getting away from near-optimal solutions (by accepting the worse solutions), the value for these solutions gets improved.

Increasing the maximum iterations in each domain causes producing multiple neighbours for each member in a constant domain. So, the algorithm searches the space more thoroughly and it is possible that better solutions are obtained. On the other hand, this incretion in each domain causes the search to take longer. Figure 2 shows a pseudo code of the MOVDO algorithm.

Start
Setting values for maximum iteration (MaxIt), Initial Population (nPop), Initial Domain (A_0), maximum iteration in each domain (L), damping coefficient (γ) and standard deviation (σ)
Generating initial population P and t=1
Assessing initial population
Performing non-dominant sorting (FNDS) and calculating ranks
Calculating the crowding distance (CD)
Sorting the population based on the CD and ranks
1. Do while the stopping condition is not met
 2. Repeat for each particle ($X \in P$)
  3. Let l=1 and Do
  4. Create a neighbourhood (Y) and assess it
   5. If Y dominates X, let X=Y; otherwise, go to the next step
    6. Randomly select a number from (0, 1), if it is less than a specific number, let X=Y; otherwise, go to next step
   7. If l=L then go to next step; otherwise, l=l+1 and go to Step 4
 8. Performing the FNDS and calculating ranks
 9. Calculating the CD
 10. Sorting the population based on the CD and ranks
 11. Updating the domain and t=t+1
12. If t=$MaxIt$, then go to the next step; otherwise, go to Step 2
Show the first Pareto frontier
End

Fig 2. Pseudo code for the MOVDO algorithm

4-2- Main steps in the MOVDO algorithm

Solution representation

- Section 1 of the solution representation

This part of the solution representation is of I*S dimension and in each scenario shows which warehouses are deployed. This matrix is filled by 0 and 1 elements. Figure 3 shows an example of this matrix.

	S1	S2
I1	0	1
I2	1	0
I3	1	1
I4	1	0

Fig 3. Example of the first section of a solution representation

- Section 2 of the solution representation

This part of the solution representation is of (K*I)*J dimension and shows the flow of product k from i to j. This matrix is filled with values between 1 and D_{jks}. If we have K=2, I=4 and J=3, an example of this matrix is shown in Figure 4.

		J1	J2	J3
K1	I1	12	8	10
	I2	18	14	9
	I3	19	10	2
	I4	7	20	23
K2	I1	16	7	9
	I2	17	10	13
	I3	9	18	3
	I4	19	18	22

Fig 4. Example of the second section of a solution representation

- Section 3 of the solution representation

This part of the solution representation shows the pre-stored value of product k in the i-th warehouse in the s-th scenario, and is of $(K*I)S$. This matrix is filled with integer values. If we have K=2, I=4 and S=3, an example of this matrix is shown in Figure 5.

		S1	S2	S3
K1	I1	3	8	4
	I2	1	4	3
	I3	6	5	7
	I4	4	6	3
K2	I1	8	9	1
	I2	3	4	6
	I3	8	2	2
	I4	5	9	1

Fig 5. Example of the third section of the solution representation

Neighbourhood structure

Two common modes of neighbourhood are swap and reversion. In this paper, both of them are used and each time one of them is randomly selected and applied to the solution representation. An example of this method is shown in Figures 6 and 7. In the swap operator for neighbourhood, two genes of a given chromosome are selected and replaced with each other (Figure 6). In the reversion operator for neighbourhood, two genes of a given chromosome are selected, then values between two genes are selected and arranged from right to left (Figure 7).

Parent	2	1	3	6	4	5
Offspring	2	4	3	6	1	5

Figure 6. Swap \ replace operator for children \ solutions mutation

Parent	2	1	3	6	4	5
Offspring	2	5	4	6	3	1

Fig 7. Reversion operator for children \ solutions mutation

Fitness function

The fitness function is just criterion for guiding the algorithm while searching for proper solutions. Commonly, it is directly derived from the objective function and defined for assessing the chromosome value. The fitness value is stored to be used in the selection method. In this research, objective functions are fitness functions of the algorithm. For the constraint of the maximum auxiliary available product stored in warehouse, the maximum unmet demand is used. In addition, the second objective function, a penalty function is used.

Stopping condition

In the proposed MOVDO algorithm, the stopping condition is a specified number of iterations. This number and population size is provided algorithm parameters.

Parameter tuning

In this section, the parameters of the MOVDO algorithm are tuned by the Taguchi method and related calculations are shown in the appendix. In this section, criteria of the efficiency comparison for multi-objective methods are also described.

5- Computational results

After explaining the algorithms, the solution representation, mechanism and structure of neighbourhood, 10 examples with different dimensions are solved using the MOVDO algorithm and GAMS software to investigate the algorithm performance. For each example, the results of the defined criteria and other information are reported and relevant analysis is performed. Since there is no real or library data in hand for the given problem; then, the required data is randomly generated.

Table 1 shows the properties and dimensions of sample problems. As it can be seen, the dimensions of the given problems are gradually increased to investigate the algorithm efficiency in solving large-sized problems.

Table 1. Properties of sample problems

No.	*I*	*J*	*K*	*S*
1	2	10	2	3
2	3	15	2	3
3	3	20	2	3
4	4	25	3	4
5	5	30	3	4
6	6	35	3	4
7	10	45	5	5
8	14	55	6	5
9	18	65	7	6
10	22	75	8	6

In table 2, the computational results for solving 10 sample problems by the MOVDO algorithm and GAMS are shown and for each problem five criteria explained in the beginning of the parameter tuning section are calculated for assessing the algorithm performance. It should be noted that the solution methods are coded by MATLAB R2016b and run on computers with Core i5 2.4 GHz CPU and 8 Gigabyte RAM. In GAMS, the solving time is set to 3600 seconds. The parameters value for the MOVDO algorithm is set based on the values determined in the parameter tuning section. Based on the tables and figures represented so far and calculated criteria, it is obvious that MOVDO has advantages in diversity, spacing, time and NOS (number of Pareto-optimal solutions) over GAMS software. In solving small-sized problems, GAMS takes less time, has less MID and higher quality compared to MOVDO. However, as the size of problems increases, GAMS loses its ability to find quality solutions in reasonably computational time.

Table 2. Computational results for 10 test problems for MOVDO and GAMS.

Test No.	**MOVDO**					**GAMS**				
	MID	**Spacing**	**Diversity**	**NOS**	**Time(s)**	**MID**	**Spacing**	**Diversity**	**NOS**	**Time(s)**
1	171896	4/04	3250	4	26	170571	9/15	2516	4	2
2	273974	70	723	7	27	273509	95	245	3	21
3	503269	632	15845	11	29	502902	888	9880	6	36
4	541940	491	9112	16	36	539858	537	4568	10	120
5	818615	723	20435	21	39	811793	788	7283	5	478
6	1143389	448	7308	8	50	1138424	504	4176	5	1780
7	1575252	469	8829	6	68	-	-	-	-	-
8	1779454	83	6694	15	83	-	-	-	-	-
9	2793858	1242	4321	7	106	-	-	-	-	-
10	2514033	180	7407	14	133	-	-	-	-	-

5-1- Statistical comparison of MOVDO and GAMS

The following charts show the comparison of the results in the given criteria. As seen in figure 8, there is no significant difference in the MID between the two methods and in some cases GAMS has a little advantage over MOVDO. On the other hand, the maximum and minimum gaps between two methods are 0.77 and 0.7 percent, respectively. In the case of small-sized problems, in which GAMS reaches the optimum solution, the quality is higher than the MOVDO algorithm. As mentioned before, lower values for the spacing metric are desirable. Figure 9 shows no significant difference between two methods. The maximum and minimum gaps between two methods are 8 and 55 percent, respectively. It shows better results for MOVDO compared to GAMS.

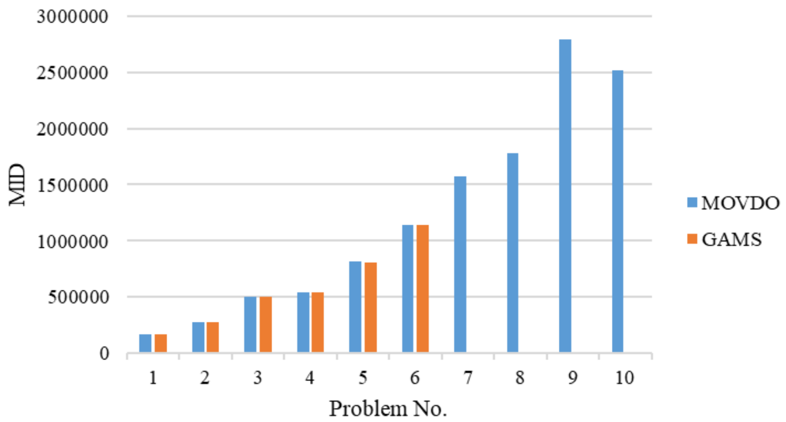

Fig 8. Comparing the MID index between MOVDO and GAMS

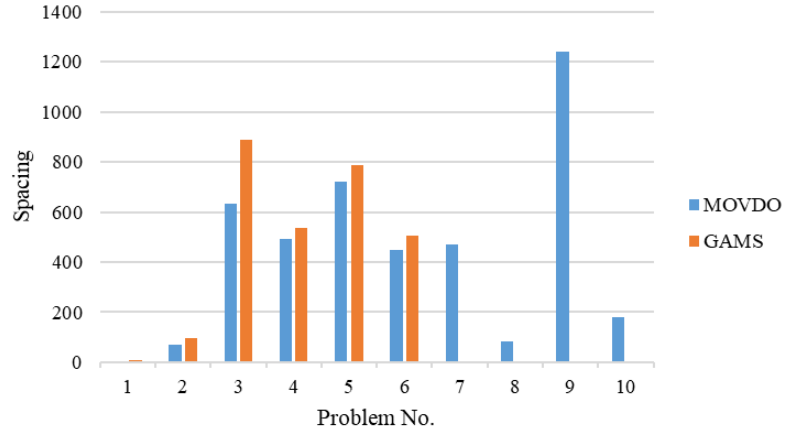

Fig 9. Comparing the spacing index between MOVDO and GAMS

As shown in figure 10, in terms of the diversity metric, MOVDO shows a better performance than GAMS. As stated before, higher values for this metric are desirable. The maximum and minimum gaps between two methods are 29 and 195 percent, respectively. They are significant values and it is seen that in the case of small-sized problems MOVDO shows a proper performance.

In figure 11, we can see that the MOVDO algorithm finds more Pareto solutions and gives more freedom to the decision maker. According to figure 12, in solving small-sized problems, MOVDO needs much less time. Of course, GAMS has very low time in solving small-sized problems and this software is not able to solve large-sized problems in reasonably computational time and the time to solve exponentially increases with the problem size. For such problems, the MOVDO algorithm has the ability to produce proper solutions in computational time. Additionally, the analysis of variance (ANOVA) with the 95% confidence level is used to give a more precise analysis about the results obtained from the proposed MOVDO algorithm and GAMS. The objective of the ANOVA is to compare the mean of several populations to determine whether the means are equal or there is significant difference between them. Regarding to the two solution methods and five criteria (or metrics), the results are mutually compared. The results obtained from this test by using MINITAB 16 are shown in tables 3 to 7.

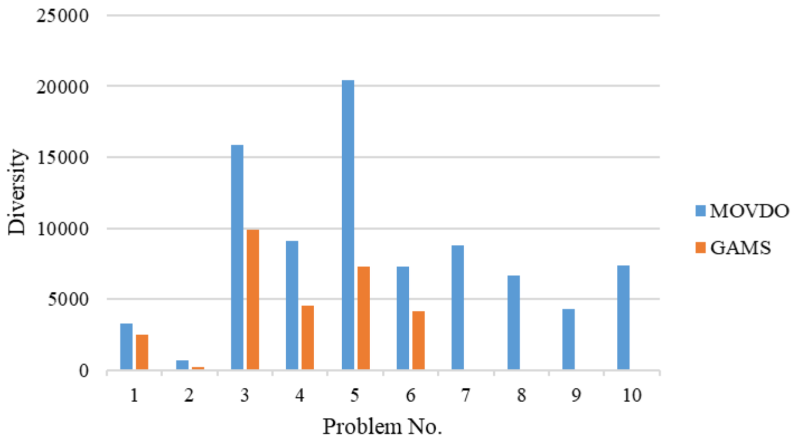

Fig 10. Comparing the diversity index between MOVDO and GAMS

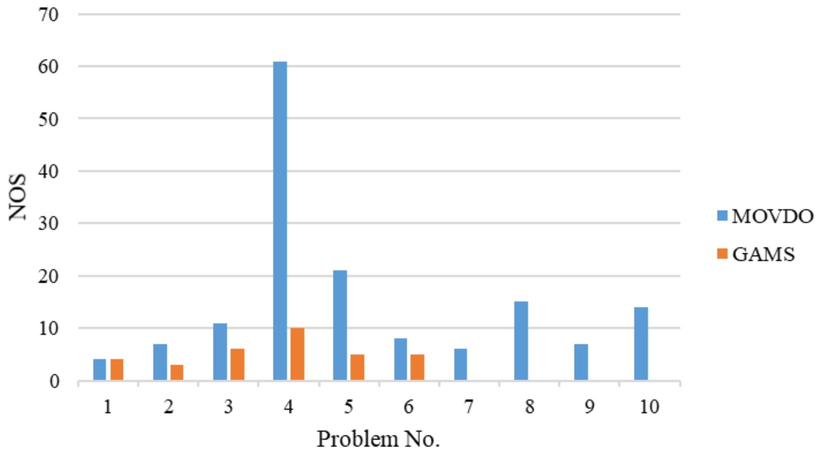

Fig 11. Comparing NOS index between MOVDO and GAMS

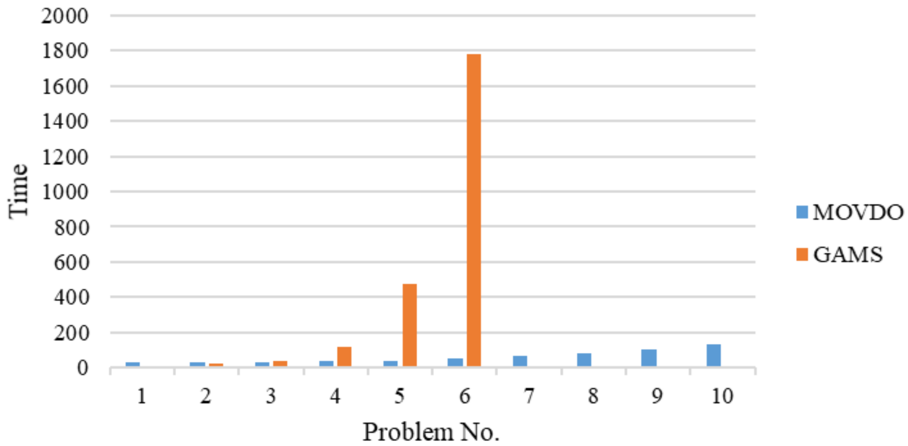

Fig 12. Comparing Time index between MOVDO and GAMS

Table 3. Output of the ANOVA for MOVDO and GAMS regarding the MID metric

Source	*DF*	*SS*	*MS*	*F*	*P*
Factor	1	1918918861	1918918861	0	0.962
Error	18	1.49100e+13	8.28334e+11		
Total	19	1.49119e+13			

Table 4. Output of the ANOVA for MOVDO and GAMS regarding the spacing metric

Source	*DF*	*SS*	*MS*	*F*	*P*
Factor	1	6702	6702	0/04	0.848
Error	18	3210882	178382		
Total	19	3217584			

Table 5. Output of the ANOVA for MOVDO and GAMS regarding the diversity metric

Source	*DF*	*SS*	*MS*	*F*	*P*
Factor	1	83199363	83199363	3/87	0.045
Error	18	386765122	21486951		
Total	19	469964485			

Table 6. Output of the ANOVA for MOVDO and GAMS regarding the NOS metric

Source	*DF*	*SS*	*MS*	*F*	*P*
Factor	1	125	125	8/65	0.009
Error	18	260	14/46		
Total	19	385/2			

Table 7. Output of the ANOVA for MOVDO and GAMS regarding the time metric

Source	*DF*	*SS*	*MS*	*F*	*P*
Factor	1	30811	30811	3	0/1
Error	18	184738	10263		
Total	19	215549			

Based on the *p*-value obtained for each metric we can analyse the results. Regarding the MID metric, the *p*-value for the both algorithms are more than 0.05, which shows that there is no significant difference. Based on the computational results given in table 2, it can be seen that the performance of MOVDO is a higher (i.e., MID is less for it). In comparison of the spacing metric between the MOVDO algorithm and GAMS, there is no significant difference and the *p*-value is more than 0.05. It is also observed that MOVDO shows a better performance in this metric compared to GAMS. For diversity, NOS, and time metrics, the *p*-value is less than 0.05, so there is a significant difference between the two methods. Thus, the MOVDO algorithm has a better efficiency.

6- Conclusion

In this paper, we tried to design a robust supply chain network employing the chance-constraint and failure matrix approaches. A multi-commodities scenario-based supply chain model was proposed to minimize the probability of the exceeded travel time at a pre-specific threshold value. The obtained results showed that a 75-node network could be solved. Applying other robust approaches (e.g., minimax regret) and employing other solution methods (e.g., Bender's decomposition) can be considered for future research.

References

Amiri, A. (2006). Designing a distribution network in a supply chain system: Formulation and efficient solution procedure. *European journal of operational research*, *171*(2), 567-576.

Anaraki, A. Z., Jeihoonian, M., Pazoki, M., & Navaei, J. A BENDERS DECOMOSITION APPROACH FOR A PRODUCTION-DISTRIBUTION NETWORK DESIGN PROBLEM.

Dasci, A., & Verter, V. (2001). A continuous model for production–distribution system design. *European Journal of Operational Research*, *129*(2), 287-298.

Hajipour, V., Mehdizadeh, E., & Tavakkoli-Moghaddam, R. (2014). A novel Pareto-based multi-objective vibration damping optimization algorithm to solve multi-objective optimization problems. *Scientia Iranica. Transaction E, Industrial Engineering*, *21*(6), 2368.

Hajipour, V., Mehdizadeh, E., & Hinojosa, Y., Kalcsics, J., Nickel, S., Puerto, J., & Velten, S. (2008). Dynamic supply chain design with inventory. *Computers & Operations Research*, *35*(2), 373-391.

Kannan, G., Sasikumar, P., & Devika, K. (2010). A genetic algorithm approach for solving a closed loop supply chain model: A case of battery recycling. *Applied Mathematical Modelling*, *34*(3), 655-670.

Mastrocinque, E., Yuce, B., Lambiase, A., & Packianather, M. S. (2013). A multi-objective optimization for supply chain network using the bees algorithm. *International Journal of Engineering Business Management*, *5*(Godište 2013), 5-38.

Mehdizadeh, E., & Tavakkoli-Moghaddam, R. (2009, May). Vibration damping optimization algorithm for an identical parallel machine scheduling problem. In *Proceeding of the 2nd International Conference of Iranian Operations Research Society, Babolsar, Iran*.

Melo, M. T., Nickel, S., & Saldanha-Da-Gama, F. (2009). Facility location and supply chain management–A review. *European journal of operational research*, *196*(2), 401-412.

Melo, M. T., Nickel, S., & Saldanha da Gama, F. (2003). Largescale models for dynamic multicommodity capacitated facility location.

Mohammadi, M., Jolai, F., & Tavakkoli-Moghaddam, R. (2013). Solving a new stochastic multi-mode p-hub covering location problem considering risk by a novel multi-objective algorithm. *Applied Mathematical Modelling*, *37*(24), 10053-10073.

Mokashi, S. D., & Kokossis, A. C. (2003). Application of dispersion algorithms to supply chain optimisation. *Computers & chemical engineering*, *27*(7), 927-949.

Nozick, L. K. (2001). The fixed charge facility location problem with coverage restrictions. *Transportation Research Part E: Logistics and Transportation Review*, *37*(4), 281-296.

Pan, F., & Nagi, R. (2013). Multi-echelon supply chain network design in agile manufacturing. *Omega*, *41*(6), 969-983.

Pishvaee, M. S., Rabbani, M., & Torabi, S. A. (2011). A robust optimization approach to closed-loop supply chain network design under uncertainty. *Applied Mathematical Modelling*, *35*(2), 637-649.

Pishvaee, M. S., Farahani, R. Z., & Dullaert, W. (2010). A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Computers & operations research*, *37*(6), 1100-1112.

.

Rabbani, M., & Yousefnejad, H. (2013). A novel approach for solving a capacitated location allocation problem. *International Journal of Industrial Engineering Computations*, *4*(2), 203-214.

Syarif, A., Yun, Y., & Gen, M. (2002). Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach. *Computers & Industrial Engineering*, *43*(1-2), 299-314.

Thanh, P. N., Bostel, N., & Péton, O. (2008). A dynamic model for facility location in the design of complex supply chains. *International journal of production economics*, *113*(2), 678-693.

Zhou, G., Min, H., & Gen, M. (2002). The balanced allocation of customers to multiple distribution centers in the supply chain network: a genetic algorithm approach. *Computers & Industrial Engineering*, *43*(1-2), 251-261.