Available online at https://www.asjp.cerist.dz/en/PresentationRevue/134

RIST

Information Processing at the Digital Age Journal

Organization of Knowledge and Advanced Technologies (OCTA)

https://multiconference-octa.loria.fr/

Chapter 3: Meta-heuristic algorithms for the multi-item transshipment problem

Noomen Selmi[a], Mohamed Hmiden[a], Lamjed Ben Said[a]*

[a]*SMART Laboratory ISG University of Tunis, 41 Liberty Avenue Bouchoucha, Tunis 2000, Tunisia*

Abstract

Differential Evolution (DE) and the Particle Swarm Optimization (PSO) are two evolutionary algorithms that confirmed their efficiency in resolving complex problems. In this paper, we intend to adopt these algorithms to resolve a complex inventory management problem, known in the literature by the transshipment problem. This problem concerns network of collaborative retailers selling items and they collaborate by exchanging items between them. The transshipment problem consists in deriving the optimal replenishment quantity, for each retailer, while a transshipment policy is adopted. A huge body of literature works has addressed this problem where several configurations are investigated. A few of them has addressed the multi-item and the multi-location configuration because of its complexity. We focus in this paper on this complex configuration and we resolve it by the PSO and DE algorithms. Secondly, we compare between the performances of these algorithms according to a set of criteria. Thirdly, we analysis the impact of the studied transshipment parameters on the inventory system performance measures.

* Corresponding author. Tel.: +216 98 509 523.
E-mail address: noomen.selmi@gmail.com.

Keywords: multi-item, multi-location; transshipment problem; Particle swarm optimization; Differential Evolution;

1. Introduction

Inventory management aims to satisfy demands while optimizing inventory performances which are generally expressed in terms of profit or cost functions. When companies evolve in an uncertain and competitive environment, this objective becomes hard to be achieved. Consequently, many inventory flexibility techniques are practiced in order to cope with these constraints and they help companies to achieve their primary objective. The substitution and the transshipment are two well-known inventory flexibility techniques largely practiced in industry. The first consists to replace the unavailable items by alternatives ones having the same functionalities. However, the second consists to transfer the items from locations in excess to ones in need. These two flexibility techniques help companies to improve their fill rate and to reduce, simultaneously, their inventory cost. The transshipment is hugely practiced in many domains as spare parts and fashion items sold in several locations. Generally, these kinds of items are replenished from suppliers, where lead times are expressed in terms of weeks or months, and transshipment is practiced to serve customers requiring, from a location, items which are out of stock. For example, when a customer looks for a specific item from ‘Zen la Soukra’ and this item is out of stock in this location, then the required item could be transshipped from ‘Zen Manar 2’ where the item is in excess. So, the customer demand is satisfied, the profit in ‘Zen la Soukra’ is improved and the inventory cost of ’Zen Manar 2’ is reduced. The transshipment problem has been studied since 1965 where several configurations, parameters and approaches are investigated. The multi-item transshipment variant is considered as a complex problem and there are a few of works that treated it. Generally, these works focused mainly on the two-location or the two-item configurations and they looked for a transshipment policy for all items simultaneously. In this paper, we focused on the multi-location and the multi-item configuration.

Our contributions here are threshold: first we studied the multi-location and multi-item transshipment problem considering periodic review and we formulate the studied problem, second we resolve the problem with PSO and DE and we compare between their performances and we study the impact of transshipment and uncertainty on the inventory system performance.

The rest of the paper is organized as follows: the second section presents a literature review of the transshipment. in the third section, we present the studied problem and its formal model. The fourth section is dedicated to the meta-heuristics algorithms PSO and DE and their application to resolve the studied problem and finally the fifth section is related to experimentation.

2. Literature review

The transshipment problem consists to derive the optimal replenishment quantities where transshipment policy is adopted, has been studied. So, many configurations, parameters and approaches are investigated, and many transshipment policies are identified. Paterson et al., 2011 overviewed works related to transshipment problem and they proposed a classification based on a set of criteria linked to: replenishment parameters, transshipment policies and environmental parameters. They identified two classes of works according to the transshipment policies: the reactive transshipment and the proactive one in Seidscher et al., 2013. The reactive is triggered once the demands are observed and the locations in need and other in excess are identified. In contrast, the proactive transshipment is started before the realization of demand and it aims to redistribute

stock in order to avoid a possible shortage. Two streams of works are identified according to number of locations involved: the two-location and the multi-location transshipment problem. The first stream adopts exact methods in order to derive the optimal inventory decisions. Krishnan and Rao, 1965 are the first that studied the two-location transshipment problem. They developed a single period model aiming to minimize an inventory cost function expressed in terms of holding and shortage costs. Since that, many research's focused on this stream and investigated many configurations. Among recent ones, we quote Olsson F. in 2015, who considered the transshipment lead times for two-location inventory system adopting continuous reviewing. Yao et al., 2016 studied the two-location transshipment problem with a single replenishment, at the beginning of a season, while reactive transshipment is practiced during the season. Feng et al., 2019 studied the two-location transshipment problem in a competitive context and with dynamic demand information's. The second stream of works focused on the multi-location configuration and it adopts simulation-based methods to derive approximate solutions proposed by KÖCHEL, P.,1998 and Kochel, P. et al., 2005. The meta-heuristic is a simulation-based optimization approach which is widely adopted to resolve complex problem as the multi-location problem considering uncertain demands. Miao Z., 2008 resolve the transshipment problem with fixed schedules with a genetic algorithm. Hochmuth and kochel, 2012 resolve the multi-location transshipment problem with many realistic parameters with particle swarm optimization (PSO) Algorithm. Danloup et al., 2018 compared the performances of two meta-heuristics applied for the transshipment problem: Local Neighborhood Search and genetic algorithm. All the mentioned works above, focused mainly on the single item configuration. Few of works have interested in the multi-item configuration. They investigated in the two-location network with periodic reviewing or the multi-location configuration with continuous reviewing.

In this paper, we focus on the transshipment problem for multi-location and multi-item configuration considering uncertain demands. We aim to resolve the studied problem with meta-heuristic approach. We compare between the performances of two algorithms, Particle swarm optimization (PSO) and Differential Evolution (DE), according to a set of criteria.

3. Problem description

We study an inventory system composed of N locations selling many products. At the beginning of the period, these locations are replenished from a common supplier and over the period demands are observed and satisfied. At the end of the period, a location Li could be in excess related to product Pk and in need for Pm. However, location Lj could be in need for Pk and in excess for Pm. Transshipment, from Locations Li to Lj of Pk units, corrects Pk shortage at Lj and it reduces the Pk holding cost at Li. Here, we consider the fixed transshipment cost. The goal is to determine the transshipped quantities between locations and the replenishment quantities of products at each location optimizing a profit function. In order to introduce our studied problem, we present an illustrative example, shown by Figure 1, of three locations L1, L2 and L3 selling two products P1 and P2. P1 is in need (-3) at L1, in excess (+4) at L2 and in need (-2) at L3. However, P2 is in excess (+4) at L1, in excess (+3) at L2 and in need (-6) at L3. Shortages at L1 and L3 could be corrected by transshipping items from L1 (P1) and L2 (P1, P2).

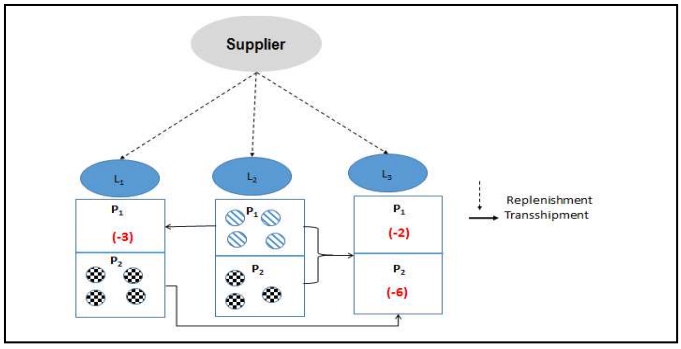


Fig. 1. Illustrative example of three locations

3.1. Notations

Throughout this paper, we adopt the following notations

- *Indexes:*

i index of location $i \in \{1, 2, \ldots, N\}$.

k index of product, $k \in \{1, 2, 3, \ldots, K\}$.

- *Parameters:*

P_i^k The unit selling price of the product k at location i.

r_i^k The shortage unit cost of product k at location i.

s_i^k The salvage unit cost of product j at location i .

c_i^k The replenishment unit cost of product k at location i .

tc_{ij}^k The transshipment unit cost between locations i *and* j for product k

D_i^k The demand of the **product k at location i.**

$f(D_i^k)$ The probability density function of the demand of the **product k at location i.**

- *Performance measure functions*

$\pi(X, D)$ The total profit function

$TC(X, D)$ The total Inventory cost

$TR(X, D)$ The total revenue generated before transshipment execution.

$TP(X, D)$ The total transshipment profit

3.2. Problem formulation

$$\pi(X,D) = \ TR(X,D) - TC(X,D) \ + \ TP(X,D) \tag{1}$$

$$TR(X,D) = \sum_{i=1}^{N}\sum_{k=1}^{K} p_i^k \times \min\{D_i^k, Y_i^k\} \tag{2}$$

$$TC(X,D) = \sum_{i=1}^{N}\sum_{k=1}^{K} c_i^k \times Y_i^k + \sum_{i=1}^{N}\sum_{k=1}^{K} s_i^k \times \left(Y_i^k - D_i^k\right)^+ + \sum_{i=1}^{N}\sum_{k=1}^{K} r_i^k \times \left(D_i^k - Y_i^k\right)^+ \tag{3}$$

$$TP(X,D) = \sum_{i=1}^{N}\sum_{\substack{j=1 \\ i\neq j}}^{N}\left(\sum_{k=1}^{K}\left(s_i^k + P_j^k \ - tc_{ij}^k\right) \times T_{ij}^k\right) \tag{4}$$

$$\sum_{i=1}^{N} T_{ij}^k \leq \left(D_j^k - Y_j^k\right)^+ \quad \forall\, j \in \{1,2,\dots,N\}, \forall\, k \in \{1,2,\dots,K\} \tag{5}$$

$$\sum_{j=1}^{N} T_{ij}^k \leq \left(Y_i^k - D_i^k\right)^+ \quad \forall\, i \in \{1,2,\dots,N\}, \forall\, k \in \{1,2,\dots,K\} \tag{6}$$

4. DE and PSO for the Transshipment problem

Here, we resolve the studied problem descripted and formulated above by two Evolutionary Algorithms, Differential Evolution (DE) and Particle Swarm Optimization (PSO), that confirmed their efficiency in resolving complex problem.

4.1. Differential Evolution

Differential Evolution (DE) was introduced for the first time by Storn and Price, 1997, as a stochastic and population-based optimization algorithm. DE was proved to be the fastest evolutionary algorithm (EA) and it was used for solving nonlinear optimization problem over continuous spaces. DE has been shown having a good convergence and very simple but very powerful for optimizing continuous functions. As many others evolutionary algorithms, DE uses three operations: mutation, crossover and selection that guide the individuals of (population) to move toward a global optimum. DE was used to solve diverse optimization problems and it has proved his performance. The DE algorithm results depends enormously of the mutation strategy and the control parameters: the population size (NP), the crossover operator (CR) and the differential weight factor (F).

The general structure of DE algorithm is composed by the steps shown in figure 2. The steps are executed sequentially till stop continuation is met. These steps are:

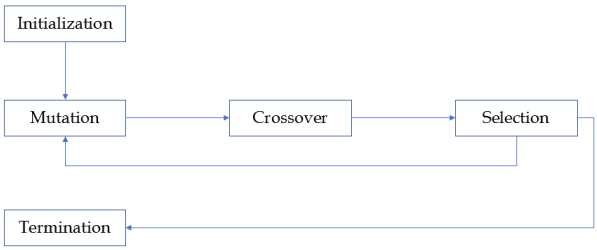


Fig. 2. DE general structure

- Initialization: DE starts with initialization of the population by producing NP individuals in the problem space domain. Each individual Xi is presented by a vector of D values (each value associated to one variable of the problem dimension):

$X_i = \{x_i^1, x_i^2, \ldots.., x_i^N\}$, i ∈ {1, 2, …., NP}, where N is the number of locations.

$x_i^j = \left(x_i^{|1}, x_i^{|2}, \ldots, x_i^{|K}\right)$, where k is the number of items.

- Mutation: After the initialization and in each generation g, DE uses the mutation operation to create at mutant vector Vi,g associated to each target vector Xi,g (Xi at generation g), $V_{i,g}=\{\mathbf{v}_{i,g}^1, \mathbf{v}_{i,g}^2, \ldots.., \mathbf{v}_{i,g}^D\}$ with i ∈ {1, 2, …., NP} and g ∈ {1, 2, …., G}, (G is the maximum number of generation. There are different DE variants identified according to mutations operation formula. Here, we are limited to the DE best/1 having the following mutation formula:

$$V_{i,G} = X_{best,G} + F * \left(X_{r1,G} - X_{r2,G}\right) \tag{7}$$

- Crossover operation: The crossover step, as presented by the formula below, is used to introduce some diversity in the population during each generation in order to look for the optimum. In this phase DE produce, at each generation g, a trial vector $U_{i,g}= \{\mathbf{u}_{i,g}^1, \mathbf{u}_{i,g}^2, \ldots.., \mathbf{u}_{i,g}^D\}$ associated to each individual Xi,g. The binomial Version of the crossover operation is presented below:

$$u_{i,g}^j = \begin{cases} v_{i,g}^j & if\ rand < CR\ or\ j == Rand \\ x_{i,g}^1 & otherwise \end{cases} \tag{8}$$

CR represents the crossover parameter (CR ∈ [0.1]), Rand is a random integer value with Rand∈ [0.D].

- Selection: During this step, at the generation g DE have to decide which vector to keep between the pair Xg and Ug depending on their fitness values, for the generation g+1.

$$X_{g+1} = \begin{cases} U_g & if\ fitness(U_g) > fitness(X_g) \\ X_g & otherwise \end{cases} \tag{9}$$

This selection formula is in case of maximization problem.

The algorithm restarts the cycle from the mutation phase until the stop condition is reached.

The main DE parameters are:

- The Cross-over probability (CR).
- The differential weight factor (*F*)
- The population size (NP).

Table 1. DE algorithm

DE algorithm (BEST/1)
1. P (NP) ← Initialize Population
2. Evaluate each individual of P by **algorithm 2**
3. **While** (Stop-condition not met) Do
4. **For** i= 1 to NP
5. Radom choose {$individual_1$ and $individual_2$} from P
6. $Individual_{best}$ ←Look for the best individual from P
7. R ← RandomInteger [0, N*k]
8. **For** j = 1 to (N*K)
a. CrossoverProbability ← random()
b. **If** (crossoverProbability < CR or R== j)
$Candicate_j$ ← $Individual_{best,j}$ + W * ($individual_{1,j}$ – $individual_{2,j}$)
c. Else
$Candidate_j$← $individual_{i,j}$
9. **End for**
10. Evaluate candidate fitness by algorithm 2
If (Candidate Fitness > $Individual_i$ Fitness) **Then**
$Individual_i$ ← Candidate
11. **End for**
12. **End while**

4.2. PSO algorithm

The Particle swarm optimization (PSO) is one of the most known population-based algorithms proposed by Eberhart and Kennedy,1995. Since its appearance, several improvements were introduced to the basic version and was used for solving global optimization problems. The basic operation of Particle Swarm Optimization

consists in searching for the optimal solution in a search space D (number of dimension) where each particle 'i' is characterized by its position (Xi) and its velocity (Vi) which are represented as follows:

$X_i = \{x_i^1, x_i^2, \ldots\ldots, x_i^N\}$, i ∈ {1, 2, …., NP}. Where N is the number of locations.

$$X_i^j = \left(x_i^{j1}, x_i^{j2}, \ldots, x_i^{jk}\right) \quad \text{, where k is the number of items}$$

Each particle must also keep track of its best previous position $P_{besti} = (p_{i1}, p_{i2}, p_{iD})$ as the best among all the particles of the population $P_{gbest} = (p_{g1}, p_{g2}, p_{gD})$. At each iteration, each particle in the population adjust its velocity and calculate its new positions vector using the best fitness in the population according to the following two formulas:

$$v_{id} = w v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (10)$$

$$x_{id} = x_{id} + v_{id} \quad (11)$$

Where c1 and c2 are the acceleration constants and w is the inertia weight parameter. r1 and r2 are two random generated numbers in the interval [0, 1] according to the uniform law.

Table 2. PSO algorithm

PSO algorithm
1. P (NP) ← Initialize Population
2. **While** (Stop-condition not met) Do
3. Update pBest of each particle
4. Update gBest of the population
5. **For** i= 1 to NP
a. Candidate← $Particle_i$ (copy $Particle_i$ in Candidate)
b. Calculate new_Velocity for Candidate according to (10)
c. Calculate new_Location for Candidate according to (11)
d. Evaluate Candidate fitness by algorithm 2
e. **if** (Calculate Fitness > $Particle_i$ Fitness) Then
$Particle_i$ ←Candidate
6. **End for**
7. **End while**

4.3. Expected profit algorithm

Using the below expected profit algorithm, we estimate the profit generated by the system for a specific combination of a predetermined inventory vector and a set of generated demand vectors. This …….s

Table 3. Expected profit algorithm

Expected profit algorithm
1. Expected_profit←initilize_to_zero()
2. X inventory vector
3. **For** i= 1 to Number of Simulation (NS)
D← generate_Demand ()
Expected _profit← Expected _profit + $TP(X,D)$
4. **End for**
5. Expected _profit← Expected _profit /NS
6. **Return** Expected _profit

5. Experimental Results

In this experimental section, we focus on three different problem configurations of transshipment problem identified according to the number of items. We are interested in four locations selling 2, 4 or 8 items. Characteristics of the studied configurations are presented in Appendix A.

We note here, that in addition to the uniform distribution of demand mentioned in table 1, we used the normal distribution with the same parameters.

5.1. Parameters settings

The parameters of the used algorithms are tested in Noomen et al., 2020 and summarized in the Table below:

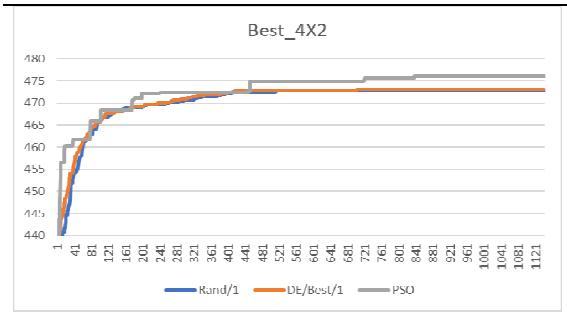
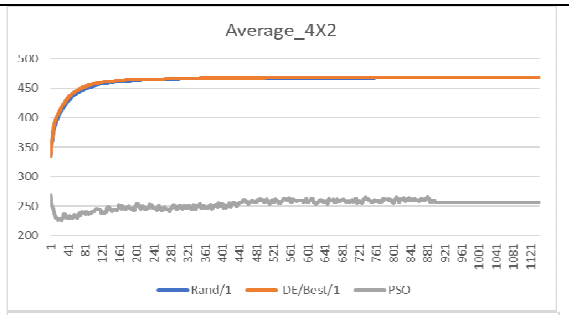
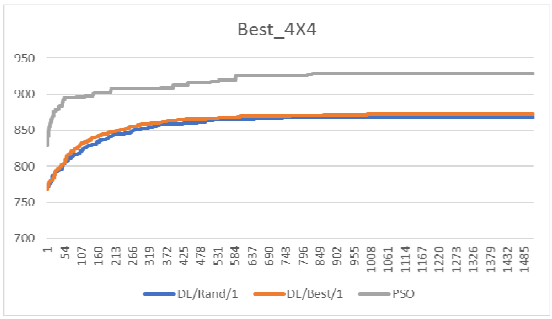
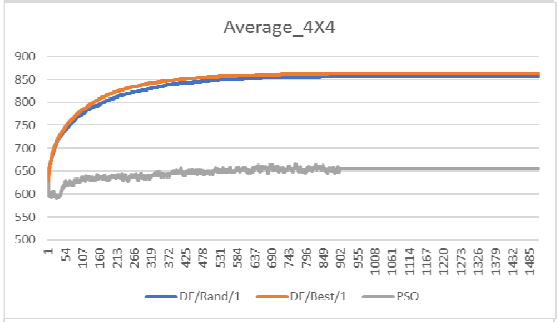
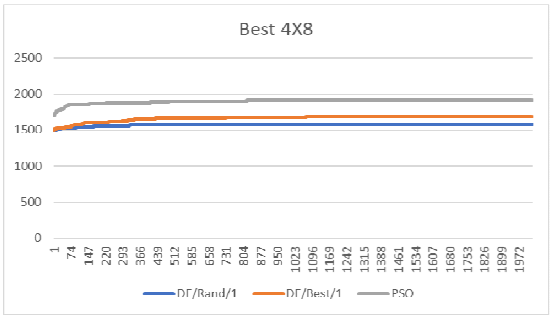
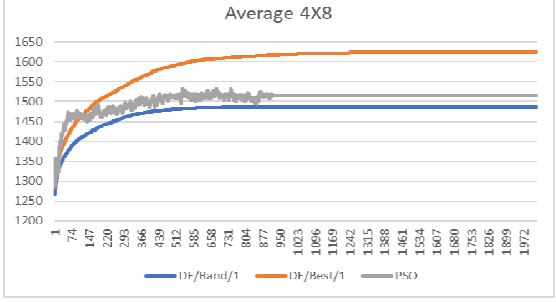
Table 4. List of used parameters for PSO and DE

DE Parameters		PSO Parameters	
Parameter	Best value	Parameter	Best value
CR	0.7	W	0.6
F	0.6	C1=C2	1.5
NP	30	NP	30*
N. generation	Stop stability condition*	N. generation	900

5.2. DEs and PSO comparison

We treated here 3 different problem configurations (3 models) and we present below the results of DE/Rand/1, DE/Best/1 and PSO showed in the table below. This table contains respectively the results of the best and the average system fitness value realized by each algorithm corresponding to the three different configurations (4X2, 4X4 and 4X8).

Table 5. DE/Rand/1, DE/Best/1 and PSO results

Best individual results	Average population results
	
	
	

The curves show that PSO always takes the top compared to DE in all models in the case of the best individual results, which proves the advance of PSO compared to DE variants. On the other hand, the average value achieved by individuals of PSO population is almost lower than the others of DE algorithms.

System performance studies: We study in this section the impact of two different parameters (fixed cost transshipment and the demand uncertainty) on the system performance. We chose also to use two demand

types distribution (uniform demand distribution and normal demand distribution). We present below the results of these experimentations

5.3. Demand Impact on system performance

We present below the results of the study of the impact of the variation of normal and uniform demand distribution on the system performance using DE/Rand/1, DE/Best/1 and PSO:

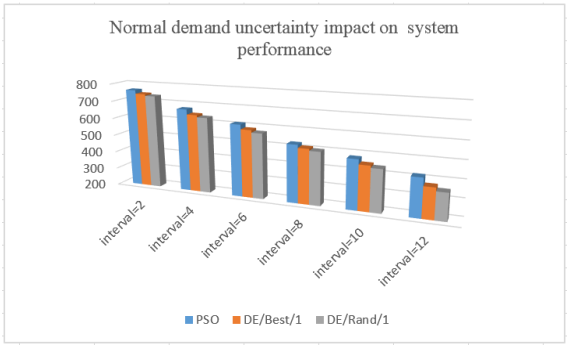


Fig. 3. Impact of normal demand on system performance

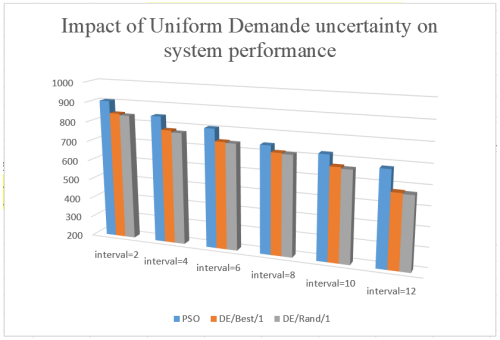
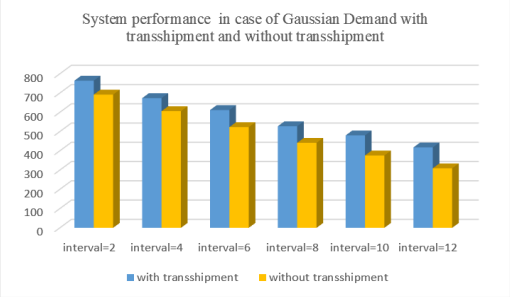


Fig. 4. Impact of uniform demand on system performance

Figure 3 and 4 show that the performance of the PSO is always better than that of the DE even with the variation of the uncertainty of the demand in both normal and uniform cases. Results shwo also that DE/Best /1 proves to be better than DE/Rand /1 in all experimentations. Our experimentations prove also that the impact of normal and uniform demand uncertainty on system performance is very remarkable; we notice also a rapid decrease of the system profit in the case of normal demand faster than in the case of uniform demand, this proves that our system depends directly on the nature of demand and its domain: The more the uncertainty of the demand increases the more it causes a decrease of the system performance and affects the result of the fitness function regardless of the algorithm.

5.4. Transshipment Impact on system performance using DE/Best/1

We present below the results of the study of the impact of the transshipment on system performance in case of normal and uniform demand distribution uncertainty variation using DE/Best/1:



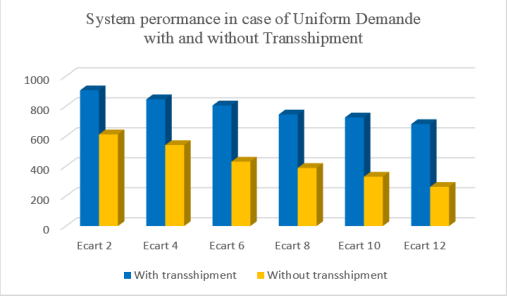
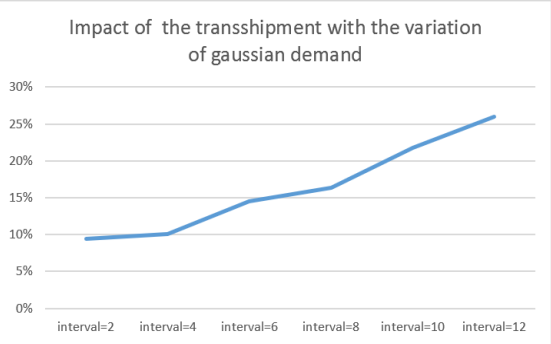


Fig. 5. *Impact of transshipment on system performance in case of normal demand*

Fig. 7. *Impact of transshipment on system performance in case of uniform demand*

Figure 6 and 7 show that the impact of transshipment, in both cases normal and uniform demand, on system performance is very remarkable; we notice also a rapid decrease of the system profit when the demand uncertainty increases. Our experimentations show also a very clear effect of the transshipment on system results which proves its added value.

We present below the gain realized by transshipment using normal and uniform demand distribution using DE/Best/1:



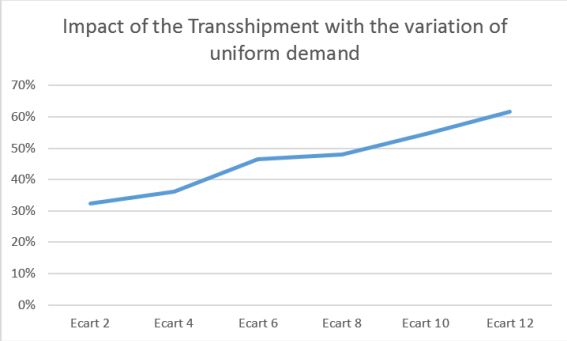


Fig. 8. *Gain made by transshipment in case of normal demand*

Fig. 9. *Gain made by transshipment in case of uniform demand*

Figure 8 and 9 show the gain realized when using transshipment, in both cases normal and uniform demand, on system performance is very important. We notice also that the gain realized in case of uniform demand is more important compared to the normal demand case.

Conclusion

In this paper, we studied the multi-location and multi-item inventory management considering lateral transshipment. we proposed two evolutionary algorithms (DE/Rand/1, DE/best/1 and PSO), to resolve the studied problem. The model considered many items which could be transshipped between locations. We proposed a simulation algorithm to derive the expected profit value of a replenishment vector. our experimental study shows that the PSO algorithm performs better results than those of DEs. We studied the uncertainty demand effect on the system performance. We noted that the transshipment impact on the system performance is more significative for high level of demand uncertainty. Our actual research could be extended by:(1) considering the fixed cost of transshipment and (2) integrating a substitution flexibility technique allowing to customer to replace their first choice by an alternative one when it is out of stock.

References

Paterson, C., Kiesmüller, G., Teunter, R., and Glazebrook, K., Inventory models with lateral transshipments: A review, European Journal of Operational Research, 210, pp. 125–136. 2011.

Seidscher, A., & Minner, S. A Semi-Markov decision problem for proactive and reactive transshipments between multiple warehouses. European Journal of Operational Research, 230 (1): 42-52. 2013.

Krishnan, K.S. and V. R. K. Rao. Inventory Control in N Warehouses. Journal of Industrial Engineering, 16, 212-215. 1965.

Olsson F. Emergency Lateral Transshipments in a Two-Location Inventory System with Positive Transshipment Leadtimes. European Journal of Operational Research. 2015;242(2):424- 433. https://doi.org/10.1016/j.ejor.2014.10.015

Yao, D., Zhou, S. X. and Zhuang, W. 'Joint initial stocking and transshipment - asymptotics and bounds', Production and Operations Management 25(2), 273–289. 2016.

Pingping Feng,Feng Wu, Richard Y. K. Fung, Tao Jia, Wei Zong. The order and transshipment decisions in a two-location inventory system with demand forecast updates. Computers & Industrial Engineering 135. DOI: 10.1016/j.cie.2019.04.043. 2019.

KÖCHEL, P. Retrospective optimisation of a two-location inventory model with lateral transshipments. In Proceedings of the 2nd International Conference on Traffic Science – ICTS '98, 129-139. 1998.

Kochel, P. & Nielander, U. Simulation-based optimisation of multi-echelon inventory systems, International Journal of Production Economics 93-94:505-513, DOI: 10.1016/j.ijpe.2004.06.046. . 2005

Miao Z., Fu K., Fei Q., Wang F. Meta-heuristic Algorithm for the Transshipment Problem with Fixed Transportation Schedules. In: Nguyen N.T., Borzemski L., Grzech A., Ali M. (eds) New Frontiers in Applied Artificial Intelligence. IEA/AIE 2008. Lecture Notes in Computer Science, vol 5027. Springer, Berlin, Heidelberg. 2008.

Noomen Selmi, Mohamed Hmiden and Lamjed Ben Said, Evolutionary algorithms for solving the multi-location and multi-item transshipment problem, OCTA conference, 2019.

Hochmuth, C. A. and K¨ochel, P.. How to order and transship in multi-location inventory systems: The simulation optimization approach. International Journal of Production Economics, 140:646–654. 2012.

N. Danloup, H. A., A comparison of two meta-heuristics for the pickup and delivery problem with transshipment. Computers & OR 2018 volume 100, 155-171.

Price, R. S. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11:341 -359. 1997.

Eberhart, R. C., & Kennedy, J.. A new optimizer using particle swarm theory. In Proceedings of the sixth international symposium on micro machine and human science(pp. 39–43), Nagoya, Japan. Piscataway: IEEE. 1995.

Appendix A.

Configurations data from Noomen et al., 2020.

Location		L1	L2	L3	L4
P1	P_i^+	15	15	14	9
	S_i^+	4	4	6	4
	r_i^+	8	5	4	5
	C_i^+	10	10	9	9
	D_j^+	u[12, 26]	u[10, 20]	u[5, 36]	u[15, 35]
P2	P_i^-	17	17	18	11
	S_i^-	5	5	5	4
	r_i^-	6	6	6	6

	C_i^-	12	12	13	11
	D_j^-	u[13, 26]	u[12, 20]	u[4, 36]	u[16, 35]
	P_i°	15	15	14	9
	S_i°	4	4	6	4
P3	r_i°	8	5	4	5
	C_i°	10	10	9	9
	D_j°	u[12, 26]	u[10, 20]	u[5, 36]	u[15, 35]
	P_i^+	17	17	18	11
	S_i^+	5	5	5	4
P4	r_i^+	6	6	6	6
	C_i^+	12	12	13	11
	D_j^+	u[13, 26]	u[12, 20]	u[4, 36]	u[16, 35]
	P_i°	15	15	14	9
	S_i°	4	4	6	4
P5	r_i°	8	5	4	5
	C_i°	10	10	9	9
	D_j°	u[12, 26]	u[10, 20]	u[5, 36]	u[15, 35]
	P_i°	17	17	18	11
	S_i°	5	5	5	4
P6	r_i°	6	6	6	6
	C_i°	12	12	13	11
	D_j°	u[13, 26]	u[12, 20]	u[4, 36]	u[16, 35]
	$P_i^\prime$	15	15	14	9
	$S_i^\prime$	4	4	6	4
P7	$r_i^\prime$	8	5	4	5
	$C_i^\prime$	10	10	9	9
	$D_j^\prime$	u[12, 26]	u[10, 20]	u[5, 36]	u[15, 35]
	P_i°	17	17	18	11
	S_i°	5	5	5	4
P8	r_i°	6	6	6	6
	C_i°	12	12	13	11
	D_j°	u[13, 26]	u[12, 20]	u[4, 36]	u[16, 35]