



HAL
open science

Bi-level programming for modelling inventory sharing in decentralized supply chains

Fatima Ezzahra Achamrah, Fouad Riane, El-Houssaine Aghezzaf

► **To cite this version:**

Fatima Ezzahra Achamrah, Fouad Riane, El-Houssaine Aghezzaf. Bi-level programming for modelling inventory sharing in decentralized supply chains. *Transportation Research Procedia*, 2022, 62, pp.517-524. 10.1016/j.trpro.2022.02.064 . hal-03668655

HAL Id: hal-03668655

<https://hal.science/hal-03668655v1>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



24th Euro Working Group on Transportation Meeting, EWGT 2021, 8-10 September 2021,
Aveiro, Portugal

Bi-level programming for modeling inventory sharing in decentralized supply chains

Fatima Ezaahra Achamrah^{a,b,*}, Fouad Riane^{a,b}, El-Houssaine Aghezzaf^{c,d}

^a*Ecole Centrale Casablanca, Ville Verte, Bouskoura 27182, Morocco*

^b*Paris-Saclay University, CentraleSupélec, Industrial Engineering Department (LGI), Rue Joliot-Curie, Gif-sur-Yvette 91190, France*

^c*Ghent University, Faculty of Engineering and Architecture, Department of Industrial Systems Engineering and Product Design, Ghent, Belgium*

^d*Industrial Systems Engineering (ISyE), Flanders Make, www.FlandersMake.be*

Abstract

This paper deals with inventory sharing and routing in the context of decentralized supply chains. The supply chain considered in this paper consists of a single manufacturer distributing its products through a network of independent Points of Sale (POS). The problem is modeled as a 1-leader - n-followers Stackelberg game. A new mixed-integer bi-level program is developed, in which the manufacturer decides first on inventory levels and the distribution routes, considering each follower's (POS) response function that minimizes the follower's own cost. A trade-off solution to manage conflict of interests between the parties involved in the supply chain is also proposed. To solve the mixed-integer bi-level program an original hybrid Genetic Algorithm coupled with deep reinforcement learning is developed and used to solve a set of large-size instances. The gap analysis shows that the proposed hybrid algorithm performs quite well and that inventory sharing allows the network to improve its service level.

© 2021 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the scientific committee of the 24th Euro Working Group on Transportation Meeting.

Keywords: Transshipment; Vehicle Routing Problem; Decentralized Supply Chain, Bi-level Optimization; Metaheuristic ; Deep Reinforcement Learning.

1. Introduction

To increase service levels and reduce logistics costs manufacturers tend to encourage, through lateral transshipment (LT) policy, sharing of inventories between a network of points of sale (POS) belonging to the same echelon (Wang et al., 2019). To successfully achieve such economies of scale, each LT policy has to balance conflicting interests and manage inventories and deliveries. Moreover, it has to choose the appropriate LT price so that both manufacturer and POS would benefit from LT and its related cost would not exceed the profit to be made (Shao et al., 2011; Atan et al.,

* Corresponding author.

E-mail address: fatimaezzahra.achamrah@centrale-casablanca.ma

2018). This is the case of decentralized supply chains in which the same manufacturer does not own POS, and each decision-maker or level of the supply chain works to optimize its costs (Li et al., 2020).

In the literature on vehicle routing (VRP) or inventory routing with LT, the problem is thoroughly studied in its centralized version (Peres et al., 2017; Lefever et al., 2018). Moreover, to simplify the optimization problems, most papers assume that LT operations are outsourced and performed by another carrier; only consider decision variables that determine nodes to visit and periods when LT may occur. However, LT often takes place with independent POS not owned by manufacturer. On decentralized decision making, LT has been extensively studied in the context of news vendor-related supply chains (Rudi et al., 2001; Hu et al., 2007; Arıkan and Silbermayr, 2018). Two approaches are used. Non-cooperative game framework in which a manufacturer acts as a Stackelberg leader (?). The other approach uses cooperative game framework (Anupindi et al., 2001; Granot and Sošić, 2003; Sošić, 2006). Authors address inventory control problems in which a set of competitive retailers define their inventories unilaterally before satisfying the demands of their customers and then collectively decide how to share their inventory. It remains, however, that the LT has been usually treated as the single simplified manufacturer multiple customers inventory management problem in which vehicle routing, delivery scheduling decision, and the relative power of all parties in the decision making are not considered.

This paper contributes to the literature in three main dimensions. First, it studies a multi-product multi-vehicle Vehicle Routing Problem (VRP) with LT and Inventory Management (VRP-TIM) in a decentralized supply chain. The supply chain consists of a manufacturer who distributes products through a network of independent POS. Demand for finite horizon planning is deterministic but time-varying. Product delivery to the POS is carried out using a homogeneous capacitated fleet of vehicles. Moreover, unlike other research, our approach considers that LT is not outsourced. That is, its related decision is integrated into the design of vehicle routing. Each player's preliminary decision is to enhance the service level while maintaining a minimum total cost, which includes transportation, inventory, lost sales, and LT. Second, the paper suggests a trade-off solution to manage conflict of interests between the supply chain's players. Indeed, as a part of the collaboration, the manufacturer and the POS may agree to incur each their own holding cost as well as a part of the cost of lost sales associated with the products shortage and a part of the cost of LT. The manufacturer incurs in turn the vehicle routing cost for regular shipments. Therefore, inventory sharing can only be attractive to all players if the LT and lost sales costs share are optimally defined. Such policies are often optimized locally as both players aim at reducing their objective functions that are narrowly defined. Therefore, solving such problems requires hierarchical decision-making, which belongs to the multilevel optimization family (?). Finally, to handle the combinatorial complexity of the model, an original hybrid Genetic Algorithm (HGA) coupled with deep reinforcement learning is proposed.

2. Problem modeling

Since the supply chain is decentralized, we formulate the problem as a bi-level program (BLP). We assume that the manufacturer acts as a Stackelberg leader (Upper Level: UL) that chooses stocking levels and routes to be built, knowing each POS's (Lower Level: LL) response function arising from minimizing its costs. In addition, we define parameter $0 \leq \epsilon \leq 1$ to denote the relative share of the cost of lost sales incurred by POS and $0 \leq \beta \leq 1$ to denote the relative share of the transshipment costs incurred by POS receiving products. The manufacturer incurs the remaining parts $(1 - \epsilon)$ and $(1 - \beta)$. The parameters ϵ and β can thus be seen as measures of the relative power of the partners in the supply chain. The related BLP is defined on a graph $G = (V, E)$ where $V = \{0, \dots, n\}$ the vertex set and $E = \{(i, j) : i, j \in V, i \neq j\}$ is the edge set. Let $V_0 = V \setminus \{0\}$ be the set of n POS (LL) and 0 be the vertex representing the manufacturer (UL). Both the LL and UL incur an inventory holding cost h_{pi} per product $p \in P = \{1, \dots, m\}$. Each has a maximum inventory holding capacity C_i . The length of the planning horizon is T with discrete periods $t \in H = \{1, \dots, T\}$. We assume that at the beginning of the planning horizon the current inventory-levels for each product p , noted I_{pi0} and expressed in terms of Stock Keeping Unit (SKU), are known for each $i \in V$. D_{pit} is the demand a point of sale $i \in V_0$ has for each period t and product p . Let $k \in K = \{1, \dots, u\}$ be the available set of homogeneous vehicles. Each vehicle has a capacity Q (in equivalent SKU), with a fixed transportation cost c_k . A distance d_{ij} (expressed in km) is associated to $(i, j) \in E$. The unit cost associated with a product p transshipped from a POS i to a POS j is a_{pij} . s_{pi} is the lost sale cost associated with product p at the level of the POS i . f_{pt} the quantity of product p produced by the manufacturer at period t . The model's notation summary is given below.

Nomenclature

- V set of nodes including the manufacturer
- P set of products indexed by p
- K set of available vehicles indexed by k
- H set of periods indexed by t

- C_i maximum inventory capacity at node $i \in V$
- I_{pi0} inventory level at the beginning of the planning horizon at each node i and product p
- D_{pi} demand to be satisfied per POS $i \in V_0$ and product p
- Q capacity of vehicle
- c_k fixed transportation cost for each vehicle k
- d_{ij} distance in km between nodes $(i, j) \in E$
- h_{pi} unit inventory holding cost for each node $i \in V$ and product p
- a_{pij} unit transshipment cost of a product p from a POS $i \in V_0$ to POS $j \in V_0$
- s_{pi} loss sales cost associated to the product p at the level of a POS $i \in V_0$
- f_{pt} quantity of product p produced at the manufacturer at period t

- I_{pit} inventory level of product p at node $i \in V$ at the end of period t
- Q_{pit} quantity of product p directly shipped from the manufacturer to node $i \in V_0$ in period t
- q_{pijkt} quantity of product p shipped from node $i \in V_0$ to node $j \in V_0$ by vehicle k in period t . It includes regular shipment from manufacturer and LT between POS
- w_{pijkt} quantity of product p transshipped from POS $i \in V_0$ to POS $j \in V_0$ by vehicle k in order to sidestep the shortage of the product p in period t
- y_{pit} lost sales quantity of product p at POS $i \in V_0$ in period t
- z_{kt} equal to 1 if the vehicle k is used in period p , 0 otherwise
- x_{ijkt} equal to 1 if the arc $(i, j) \in E$ is visited by a vehicle k in period t , 0 otherwise

The BLP for VRP-TIM can be then written as:

- Upper level:

$$\min \sum_{i \in H} \sum_{p \in P} h_{p0} I_{p0t} + \sum_{i \in H} \sum_{k \in K} c_k \sum_{i \in V} \sum_{j \in V, i \neq j} d_{ij} x_{ijkt} + (1 - \epsilon) \sum_{i \in H} \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_0, i \neq j} \sum_{p \in P} a_{pij} w_{pijkt} + (1 - \beta) \sum_{i \in H} \sum_{i \in V_0} \sum_{p \in P} s_{pi} y_{pit} \quad (1)$$

$$I_{p0t} = I_{p0t-1} - \sum_{i \in V_0} Q_{pit} + f_{pt} \quad \forall p \in P, t \in H \quad (2)$$

$$Q_{pjkt} - \sum_{k \in K} \sum_{i \in V_0, i \neq j} (w_{pijkt} - w_{pijkt}) = \sum_{k \in K} \sum_{i \in V_0, i \neq j} (q_{pijkt} - q_{pjikt}) \quad \forall p \in P, j \in V_0, t \in H \quad (3)$$

$$\sum_{p \in P} q_{pi0kt} = 0 \quad \forall i \in V_0, k \in K, t \in H \quad (4)$$

$$\sum_{p \in P} I_{p0t} \leq C_0 \quad \forall t \in H \quad (5)$$

$$\sum_{p \in P} q_{pijkt} \leq Q z_{kt} \quad \forall (i, j) \in E, k \in K, t \in H \quad (6)$$

$$\sum_{i \in V} x_{ijkt} = \sum_{i \in V} x_{jikt} \quad \forall j \in V_0, k \in K, t \in H \quad (7)$$

$$\sum_{i \in V} \sum_{k \in K} x_{ijkt} \leq 1 \quad \forall j \in V_0, k \in K, t \in H \quad (8)$$

$$\sum_{j \in V_0} x_{0jkt} = z_{kt} \quad \forall k \in K, t \in H \quad (9)$$

$$\sum_{k \in K} z_{kt} \leq u \quad \forall t \in H \quad (10)$$

- Lower level:

$$\min \sum_{i \in H} \sum_{i \in V_0} \sum_{p \in P} h_{pi} I_{pit} + \epsilon \sum_{i \in H} \sum_{k \in K} \sum_{i \in V_0} \sum_{j \in V_0, i \neq j} \sum_{p \in P} a_{pij} w_{pijkt} + \beta \sum_{i \in H} \sum_{i \in V_0} \sum_{p \in P} s_{pi} y_{pit} \quad (11)$$

$$I_{pit} = I_{pit-1} + Q_{pit} - D_{pit} - \sum_{k \in K} \sum_{j \in V_0, i \neq j} (w_{pjikt} - w_{pijkt}) + y_{pit} \quad \forall i \in V_0, p \in P, t \in H \quad (12)$$

$$\sum_{p \in P} I_{pit} \leq C_i \quad \forall i \in V_0, t \in H \quad (13)$$

$$\sum_{k \in K} \sum_{j \in V_0} w_{pijkt} \leq I_{pit-1} \quad \forall (i, j) \in E, k \in K, t \in H \quad (14)$$

The objective function (1) at the UL minimizes its total cost. The first sum corresponds to the inventory cost. The second sum is for transportation costs. The third is the total shared cost of transshipment $(1 - \epsilon)$. The last sum is the total shared cost of lost sales $(1 - \beta)$. Constraints (2) indicate the conservation conditions of inventory at the manufacturer over successive periods. Constraints (3) express the conservation of flows (inflows and outflows) at each POS j . Constraints (4) guarantee that at the end of a period, vehicles are emptied when returned to the manufacturer. Constraints (5) guarantee that inventory levels at the manufacturer do not exceed the maximal holding capacity. Constraints (6) state that the quantities transported do not exceed the vehicle capacity. Constraints (7) stipulate that when a vehicle k visits the POS j in period t , the POS j must be left in period t . Constraints (8) ensure that at the most, a POS is visited once by a vehicle k per period t . Constraints (9) stipulate that only vehicles shipping products are used. Constraints (10) indicate that only available vehicles are used. The objective function (11) at the LL minimizes the POS costs. The first sum corresponds to the inventory cost. The second is the total shared cost of transshipment (ϵ) . The last sum is the total shared cost of lost sales (β) . Constraints (12) indicate the conservation conditions of inventory at the POS over successive periods. The constraints (13) guarantee that inventory levels at each POS do not exceed the maximal holding capacity. Constraints (14) state that the quantity latterly transshipped from POS i at a period t does not exceed the initial inventory level.

2.1. Reformulation of LL model using Karush-Kuhn-Tucker conditions

The traditional idea of the reformulation of the BLP is to substitute the LL mathematical program by its Karush-Kuhn-Tucker (KKT) conditions as sufficient and necessary optimality conditions (Bouza Allende and Still, 2012). We use the Lagrangian function to compute the relative complementary conditions and dual feasibility. Accordingly, the BLP is converted into a single linear program subject to the UL feasibility constraints, LL primal and dual feasibility constraints, and complementary conditions.

3. Resolution approach

BLP problems are intrinsically hard, even for convex levels. The simplest bi-level linear programs have been proven to be strongly NP-hard (?). The complexity induced by multiple levels and/or multiple objectives makes exact approaches non-efficient to tackle large-sized problems. Solving BLP are intrinsically complex and requires hierarchical decision-making, which belongs to the multilevel optimization family (?). To be able to handle the combinatorial complexity, researchers suggest turning towards metaheuristics. Among the metaheuristic algorithms, GA has proved to be practical and quite robust in dealing with discrete problems (?Amirtaheri et al., 2017). This paper proposes a hybrid Genetic Algorithm (GA) coupled with deep reinforcement learning. Due to the limited number of pages, we limited ourselves in what follows to give a brief description of the resolution approach. A thorough description is available upon request.

3.1. Hybrid Genetic Algorithm coupled with reinforcement learning

GA is fast at global search but slow to converge (Nia and Alipouri, 2009). Furthermore, local search heuristics such as Variable Neighborhood Search (VNS) are good at fine-tuning but often fall into local optimum. In this paper, a hybrid metaheuristic that combines the properties of GA and VNS is used. Indeed, GA is used to perform a global search to escape from the local optimum, whereas VNS is used to conduct fine-tuning.

The first step consists of applying the well-known GA for the UL. An initial population of chromosomes representing the routing decisions of the UL (a sequence of POS to visit) is generated, and its related mixed-integer program is solved. A fitness that corresponds to the objective function of the UL is then computed. Later on, a double-point crossover is performed. A VNS method is used to replace any child created with its better neighbor. A reverse mutation is performed depending on the corresponding mutation rate. VNS is again applied to replace a mutated child with its better neighbor. The following population from the population size of the best available solutions is chosen. The best available solution based on the computed fitness is saved. All the steps mentioned above are repeated up until reaching a maximum number of iterations. Based on the best solution of the UL, the above steps are performed to find the best solution for the LL. Based on the best solution of the LL, an initial solution for the UL is constructed. All the steps mentioned above are repeated until no improvement of the solution is noted. All the steps mentioned above are summarized in Algorithm 1.

Algorithm 1 : HGA

- 1: Generate an initial population.
 - 2: Calculate the fitness value of each chromosome according to UP's objective function.
 - 3: Execute the crossover operator.
 - 4: Use VNS to replace any child that is generated by the crossover with its best neighbour.
 - 5: Conduct mutation process.
 - 6: Apply VNS method to substitute any child with its best neighbour generated by mutation.
 - 7: Measure the fitness value of all strengthened chromosomes within the current population; select the next population with the best possible solutions from the population size and save the best relative solution.
 - 8: Repeat steps 3 to 7 until reaching a given maximum number of iterations.
 - 9: Depending on the best UL solution, repeat the above steps for the LL.
 - 10: Depending on the best LL solution, build an initial solution for UL.
 - 11: Repeat steps 2 to 11 until the until no improvement of the solution is noted or a time limit is reached.
-

Finally, HGA, through its iterative search processes, generates a lot of data that can be turned into explicit knowledge if coupled with Machine Learning models (Talbi, 2020). This data concern solutions in the decision and the objective spaces visited during the search process, moves, elite and bad solutions. In this paper, the Deep Q-learning technique (DQ) is used to help analyze this data, extract valuable knowledge, and guide to enhance the search performance and speed of the HGA. Indeed, DQ is used to help HGA figures out precisely the best actions to perform regarding the best moves for crossover and mutation operators. With the help of a two-layer neural network, the Q-value function is approximated. The state is given as the input, and the Q-value of all possible actions is generated as the output. Once the network is trained, selecting the right action means comparing each action's possible rewards and choosing the best one.

4. Computational experiments

This section summarizes the computational experiments performed on a set of instances generated randomly by Coelho and Laporte (2013) for a multi-product inventory routing problem. The reader is referred to their paper for further details, and can be downloaded the data set from <http://www.leandro-coelho.com/instances/>. As for parameter tuning, we use the Irace package (López-Ibáñez et al., 2016). This package is implemented in R and uses Iterated Race method, which is a generalization of the Iterated F-race method for the automatic configuration of optimization algorithms. All optimization steps are carried out on a personal computer (MacBook Pro, macOS Cataline, CPU 3.3 GHz Quad-Core Intel Core i7, 8 GB of RAM). The problem is solved using the branch-and-cut solver of CPLEX 12.9 and Python 3.7, and Pytorch. For each instance, we performed ten independent runs using HGA and recorded the average of results. We refer to the instances using the following notation: [number of products] P [number of POS] N [number of periods] T.

Considering a Stackelberg game, the parameters ϵ and β can be seen as measures of the relative and bargaining power of the players in the supply chain under consideration. The objective is to determine the scenario in which LT is beneficial and the total costs are minimized. That is, determine ϵ^* the fair shares relative to LT cost given the shared parameter β of lost sales cost. To illustrate the findings and without loss of generality, we consider an instance consisting of 10 POS and five products. For a given

value of β , we vary $\epsilon \in \{0; 0.2; 0.4; 0.6; 0.8; 1\}$ and observe the variation of the total cost at the level of manufacturer and POS. The intersection of the two curves makes it possible to determine ϵ^* (see Table 1).

Table 1: Optimal shares relative to the LT for a given value of share of lost sales

β	0	0.4	0.8	1
ϵ^*	~0.95	~0.80	~0.68	~0.6

From Table 1, we can see that the manufacturer tends to become relatively more interested in instituting inventory sharing schemes, as the POS incurs higher LT and shortage costs. Indeed, whenever the manufacturer incurs a major part of the cost of lost sales, he becomes more cautious about sharing the cost of LT. And vice versa. If he incurs a large part of the transshipment cost, it is in his interest to incur a small percentage of the cost of lost sales. As a result, a fair share of costs and inventory sharing is efficient if only they make at least one party strictly better-off while making no one else worse-off.

To provide insight regarding the benefits of LT and the representativeness of the results, experiments are conducted considering two scenarios in which we compare the bi-level model for VRP-TIM with a model VRP-IM in which LT is not considered (its related variables are set to zero). Table 2 summarizes the results of comparison between CPLEX and HGA in terms of total costs (TC). They also provide the gap (GAP) computed regarding the total costs obtained using CPLEX (with a time limit of 2 hours). For each instance under consideration, we remade the same tests by varying ϵ and β in order to determine the optimal ϵ^* and β^* ; corresponding to the lowest cost recorded so far. Table 3 summarizes the results for each instance under consideration. For all instances, the breakdown of costs is provided, namely: Transportation (T), Inventory (I), Lost sales (LS), and LT (Ts). Table 3 also provides the service levels computed with regards to the satisfied and lost demands and reports CPU time in second needed to solve the models using HGA. Extra experiments have been conducted for a number of periods equal to 3 and 5, and a number of products between 1 and 50. In this paper, these results are not presented due to the limited number of pages but available upon request.

Table 2: Summary of comparison between CPLEX and HGA on small and relatively large instances- number of product varying between 1 and 5

Instance	Model	CPLEX (2hrs)	HGA	GAP (%)	Instance	Model	CPLEX (2 hrs)	HGA	GAP (%)
		TC	TC				TC	TC	
1P10N5T	VRP-IM	54906	57374	4.3	1P10N3T	VRP-IM	18516	18552	0.2
	VRP-TIM	25520	26571	4.0		VRP-TIM	9187	9233	0.5
3P10N5T	VRP-IM	83486	86809	3.8	3P10N3T	VRP-IM	29513	29810	1.0
	VRP-TIM	41948	45754	8.3		VRP-TIM	17262	17326	0.4
5P10N5T	VRP-IM	95773	100671	4.9	5P10N3T	VRP-IM	31768	31859	0.3
	VRP-TIM	64473	68463	5.8		VRP-TIM	21266	21402	0.6
1P20N5T	VRP-IM	83858	89549	6.4	1P20N3T	VRP-IM	34593	34885	0.8
	VRP-TIM	63147	63336	0.3		VRP-TIM	21355	21530	0.8
3P20N5T	VRP-IM	128805	131262	1.9	3P20N3T	VRP-IM	42537	42890	0.8
	VRP-TIM	94850	101505	6.6		VRP-TIM	30651	30950	1.0
5P20N5T	VRP-IM	150075	165037	9.1	5P20N3T	VRP-IM	57323	57775	0.8
	VRP-TIM	96008	106671	10.0		VRP-TIM	34759	34855	0.3
1P30N5T	VRP-IM	213697	214955	0.6	1P30N3T	VRP-IM	75214	75413	0.3
	VRP-TIM	126989	127369	0.3		VRP-TIM	44183	44526	0.8
3P30N5T	VRP-IM	218268	229654	5.0	3P30N3T	VRP-IM	84497	84502	0.0
	VRP-TIM	149120	165583	9.9		VRP-TIM	61422	61765	0.6
5P30N5T	VRP-IM	342772	376994	9.1	5P30N3T	VRP-IM	146285	147026	0.5
	VRP-TIM	305375	311610	2.0		VRP-TIM	120758	121877	0.9
1P40N5T	VRP-IM	698592	713216	2.1	1P40N3T	VRP-IM	298157	300525	0.8
	VRP-TIM	555475	564037	1.5		VRP-TIM	211125	213115	0.9
3P40N5T	VRP-IM	1116978	1186197	5.8	3P40N3T	VRP-IM	532164	532675	0.1
	VRP-TIM	1113083	1136364	2.0		VRP-TIM	312668	314693	0.6
5P40N5T	VRP-IM	2207908	2278178	3.1	5P40N3T	VRP-IM	1045161	1047582	0.2
	VRP-TIM	1941518	1998841	2.9		VRP-TIM	560721	564945	0.7
Average		415526	431083	4.6	Average		160045	160821	0.6

From Table 2 we can see that as expected, HGA can provide solutions with a minimum gap and with less amount of time. Regarding the benefits of LT, Table 3 shows that it is overall efficient as it reduces lost sales and thus enhances service levels. In addition, any of the involved parties can end up better off once LT is allowed. Hence, depending on the value of the parameters ϵ^* and β^* , inventory sharing may work both in favor of or against any of the players, regardless of their relative power and leadership position in the supply chain.

Table 3: Summary of results for small and relatively large instances (T=3)

Instances	Model	Level	T	I	LS	Tr	TC	ϵ^* (%)	β^* (%)	SL (%)	CPU (s)
5P10N3T	VRP-IM	UL	7750	2651	4594	0	14994	-	62	43	19
		LL	0	9344	7520	0	16865	-	62	43	19
	VRP-TIM	UL	9165	2592	178	1408	13343	40	28	74	11
		LL	0	7059	70	930	8059	40	28	74	11
5P20N3T	VRP-IM	UL	13884	6379	15996	0	36260	-	39	35	79
		LL	0	11156	10359	0	21515	-	39	35	79
	VRP-TIM	UL	13329	4863	25	2944	21160	48	96	87	76
		LL	0	10405	615	2675	13695	48	96	87	76
5P30N3T	VRP-IM	UL	30478	14134	36659	0	81270	-	45	34	37
		LL	0	35832	29924	0	65756	-	45	34	37
	VRP-TIM	UL	62146	19060	197	10028	91430	25	25	78	63
		LL	0	27109	66	3272	30447	25	25	78	63
5P40N3T	VRP-IM	UL	155834	111866	300065	0	567764	-	50	33	74
		LL	0	173799	306018	0	479818	-	50	33	74
	VRP-TIM	UL	273518	46603	343	77884	398348	19	67	86	47
		LL	0	147900	701	17996	166597	19	67	86	47

5. Conclusions and perspectives

This paper considers a deterministic, multi-product, multi-vehicle routing problem with LT; in a decentralized supply chain. The supply chain consists of a manufacturer that sells a set of products through a network of independent POS, running on a franchising scheme. We model the problem as a bi-level mixed-integer program. We consider that the manufacturer acts as a Stackelberg leader. The LL: POS optimize their objective function subject to the value of the UL variables. The manufacturer's goal is to choose a stock level and routes to be constructed (according to his objective function), knowing that the POS will follow optimally; while satisfying a set of constraints relative to vehicle routing, products availability, and inventory management. The paper also suggests a trade-off solution to manage conflict of interests between levels. Finally, it presents a novel and an original resolution approach that combines a hybrid Genetic Algorithm (HGA) coupled with deep reinforcement learning.

Computational results show that if the manufacturer agrees to take part in the costs of transshipment and lost sales, the network will improve its service level. They also highlight that the manufacturer seems increasingly more involved as more LT and POS incur shortages. Furthermore, all of the players involved can end up worse-off once LT is permitted depending on the value of ϵ^* and β^* .

To examine supply chains characterized not only by one or two decision-makers but also by many divisions at the lower level, and a mix of small and large POS arranged within a hierarchical structure, the bi-level programming problem we develop could be generalized in order to model such decentralized systems as well. Finally, as for the resolution approach, coupling HGA with reinforcement learning helps greatly enhance the quality of the feasible solutions and reduce the computational time; it would be useful to further assess their effectiveness on a real dataset and other or new variants of the problem.

References

- Amirtaheri, O., Zandieh, M., Dorri, B., 2017. A bi-level programming model for decentralized manufacturer-distributor supply chain considering cooperative advertising. *Scientia Iranica* 25.
- Anupindi, R., Bassok, Y., Zemel, E., 2001. A general framework for the study of decentralized distribution systems. *Manufacturing & Service Operations Management* 3, 349–368.
- Arikan, E., Silbermayr, L., 2018. Risk pooling via unidirectional inventory transshipments in a decentralized supply chain. *International Journal of Production Research* 56, 5593–5610.
- Atan, Z., Snyder, L.V., Wilson, G.R., 2018. Transshipment policies for systems with multiple retailers and two demand classes. *OR Spectr.* 40, 159–186.
- Bouza Allende, G., Still, G., 2012. Solving bilevel programs with the kkt-approach. *Mathematical Programming* 138.
- Coelho, L.C., Laporte, G., 2013. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research* 51, 7156–7169.
- Granot, D., Sošić, G., 2003. A three-stage model for a decentralized distribution system of retailers. *Operations research* 51, 771–784.
- Hu, X., Duenyas, I., Kapuscinski, R., 2007. Existence of coordinating transshipment prices in a two-location inventory model. *Management Science* 53, 1289–1302.
- Lefever, W., Aghezzaf, E.H., Hadj-Hamou, K., Penz, B., 2018. Analysis of an improved branch-and-cut formulation for the inventory-routing problem with transshipment. *Computers & Operations Research* 98, 137–148.

- Li, Y., Liao, Y., Hu, X., Shen, W., 2020. Lateral transshipment with partial request and random switching. *Omega* 92, 102134.
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., Stützle, T., 2016. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives* 3, 43 – 58.
- Nia, M., Alipouri, Y., 2009. Speeding Up the Genetic Algorithm Convergence Using Sequential Mutation and Circular Gene Methods.
- Peres, I.T., Repolho, H.M., Martinelli, R., Monteiro, N.J., 2017. Optimization in inventory-routing problem with planned transshipment: A case study in the retail industry. *International Journal of Production Economics* 193, 748–756.
- Rudi, N., Kapur, S., Pyke, D.F., 2001. A two-location inventory model with transshipment and local decision making. *Management science* 47, 1668–1680.
- Shao, J., Krishnan, H., McCormick, S.T., 2011. Incentives for transshipment in a supply chain with decentralized retailers. *Manufacturing & Service Operations Management* 13, 361–372.
- Sošić, G., 2006. Transshipment of inventories among retailers: Myopic vs. farsighted stability. *Management science* 52, 1493–1508.
- Talbi, E.G., 2020. Machine learning into metaheuristics: A survey and taxonomy of data-driven metaheuristics Working paper or preprint.
- Wang, Z., Cui, B., Feng, Q., Huang, B., Ren, Y., Sun, B., Yang, D., Zuo, Z., 2019. An agent-based approach for resources' joint planning in a multi-echelon inventory system considering lateral transshipment. *Comput. Ind. Eng.* 138, 106098.