American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-10, Issue-03, pp-135-145
www.ajer.org
Research Paper Open Access

Hybrid Metaheurístic to Inventory Routing Problem

Diego Moah
[1](State University of Pará, Belém, Brasil)
Rafael Martinelli
[1](Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brasil)
Isaac Pereira de Sousa
[2](Federal Rural University of the Amazon, Parauapebas, Brasil)
Corresponding Author: Isaac Pereira de Sousa

ABSTRACT : *The Inventory Routing Problem (IRP) deals with routing and inventory costs at the same time. This problem comes from the context of a Vendor Managed Inventory (VMI) system in which the vendor is responsible for managing the customer's inventory. It is the combination of transportation and inventory management problems, which correspond to the higher costs in a logistics operation This problem belongs to the class of NP-Hard problems, so this study proposes a hybrid heuristic for the problem. The inventory problem was modeled as a maximum flow problem at minimum cost and to solve it we used the Network Simplex algorithm. For the routing problem we used Random Variable Neighborhood Descent and to escape from the optimal locations we used Simulated Annealing. IRP variant considering several periods and several vehicles. Each iteration of the metaheuristic is divided into two stages: the first is modifying the position of one or more customers attended by the vehicles and periods, and a second step that solves a Maximum Flow at Minimum Cost problem, to optimally assign the load volumes transported to each customer in each vehicle in each period. Then, this approach is tested in classical instances for this IRP variant, obtaining results that prove the efficiency of the algorithm The algorithm surpassed some results of the literature and in no moment was above 2% of them.*

KEYWORDS *Inventory Routing Problem; Heuristics; Integer Programming.*

Date of Submission: 27-02-2021 Date of acceptance: 12-03-2021

I. INTRODUCTION

The Inventory Routing Problem (IRP) refers to the distribution of a specific product, from a specific Distribution Center (DC), which meets a several costumers within a previous planning. The costumer consumes the product with a determined tax and has a maximum storage capacity. The products distribution is done by the company's fleet, being heterogeneous (different capacities) or homogeneous (same capacity) then. The different costs of transportation and storage is the supplier's responsibility, and the main goal is to minimize both this costs during the whole planning, aiming to have no delivery breaks.

This problem is known as a NP-hardness problem and some authors have proposed several solutions such as: the Exact Method developed by Archetti et al. (2007); the Heuristic one developed by ALNS and proposed by Coelho et al. (2012b); to heterogeneous fleet, there are the Branch and Cut Exact Algorithms of Adulyasak et al. (2013a) e Coelho e Laporte (2013b). Coelho et al. (2014), in his work, makes a detailed revision about the algorithmic aspects of IRP.

The first heuristics to the IRP used to have many approximations and to consider some factors, such as the work of Dror et al. (1985), who considered the transportations costs to solve the problem. Nowadays, the problem has approached with more developed algorithms that search for high quality results; these metaheuristics have the capacity to flee from optimum locals and clever searches in the solution space.

Coelho (2012b) proposed a metaheuristic to the IRP, MIRP, IRPT and MIRPT. The author uses Random Variable Neighborhood Descent (RVND) and when the algorithm finds the optimum local and the improvements movements have not been allowed it is used the Simulated Annealing (SA) as a solution's acceptance criterion. The author uses neighborhood research, being 12 of them.

Archetti et al. (2012) proposed a hybrid heuristic to the IRP using a Tabu Search (TS) metaheuristic with other two MIPs to the ML's and OU's politics. The author named HAIR (Hybrid Approach to Inventory Routing) the metaheuristic. The first MIP receive the TS' results then the MIP runs the route changing among the periods do not altering the routes, but only its specific period. The second MIP removes and inserts the costumers into the routes.

Peres et al. (2017) introduces a new hybrid metaheuristic in literature to the Multi-Product Multi-Vehicle Inventory (MMIRPT), which has based in the neighborhood searches to the routing problem and in the network simplex algorithm to the transportation problem. Inspired by Coelho et al. (2012b) the proposed algorithm also used the RVND and SA as a criterion to accept solutions, aiming to flee from optimum locals when a better solution is found during the research, however it was not tested in instances. The proposed metaheuristic in this paper is inspired by Peres' work, who researched about the maximum optimal maximum flow algorithm with a minimum cost to the storage problem.

This paper is organized as follows: a brief introduction to the theme with context, motivation and goal. The Section 2 details the approached problem. In Section 3 it is described the heuristic used in this paper. The Section 4 presents the computational experiments' results. Finally, in Section 5 there are the last conclusions and suggestions for future research.

II. INVENTORY ROUTING PROBLEM FORMING

The IRP can be formulating as follows: a complete graph and non-oriented G = (V, E) so that the set of vertexes and the set of arcs are defined as V = {0, ..., n} and E = {(i, j) ∈ V, i ≠ j}, respectively. The vertex 0 belonged to V is the Distribution Center (DC) and V' = V \ 0 show the costumers. The arcs (i, j) ∈ V have a cost c_{ij} > 0. The problem regards decisions in a horizon T = {0, ..., p}. The costumer i demands d_i^t, $\forall t \in T$, has a storage maintenance cost h_i and a storage capacity L_i, DC has the storage L_0. When t = 0, it can exist, for the costumer, an initial storage defined by I_i^0, $\forall i \in V$. Considering the DC has a sufficient storage to comply de costumers' demand along T, and the DC must choose which period t will be complied the demand of the costumer i to a quantity q_i^{kt}. The fleet can be heterogeneous or homogeneous of k vehicles, K={1,...,k}, and its capacity Q_k is available in DC, in this research it has considered only 1 vehicle. Each vehicle k does a single route in each period t, visiting a specific number of costumers. The variables are described as bellow:

Table I: Sets and indexes of the Model.

SETS	SETS	INDEXES
DC	V	i=0
Nodes	V′	i, j
Vehicles	K	K
Periods	T	T

Table II: Parameters of the Model.

PARAMETERS	DESCRIPTION
h_i	Storage Cost per unit of costumer i
c_{ij}	Cost of the Edge i, j
d_i^t	Demand of the costumer i in the period t
L_i	Storage Capacity of the costumer c
Q_k	Transportation Capacity of the vehicle k

Table III: Variables of the Model.

VARIABLES	DESCRIPTION	DOMAIN
x_{ij}^{kt}	1 if the arc (i, j) is coursed in the period t by the vehicle k; 0, otherwise	{0;1}
y_i^{kt}	1 if the costumer i be visited by the vehicle k in the period t; 0, otherwise	{0;1}
I_i^t	Storage Level of the costumer i at the end of the period t	Z^+

q_i^{kt}	Transportated Quantity to costumer i in the period t by the vehicle k	Z^+
w_{ij}^{kt}	Flow to j from i in the period t	Z^+

The IRP aims to minimize the whole costs of transportation and storage, assuming the following restrictions:

- The storage I_0^t of DC must be lower or greater than L_0.
- The storage I_i^t of the costumer i must be lower or greater than L_i.
- The quantity delivered by the vehicle k in the period cannot exceed the quantity Q_k.
- Each vehicle must start and finish its route by the storage area.
- Each vehicle must do only one route by period and the number of routes cannot overcome de fleet's availability.
- Each costumer c must be visited by only one vehicle in each period.

The IRP based in Coelho (2012a) could be formulated as a model of PLIMB with the Maximum Level (ML) politic, as described below:

$$Min \sum_{i \in V} \sum_{t \in T} I_i^t h_i + \sum_{i \in V} \sum_{j \in V\, i \neq j} \sum_{k \in K} \sum_{t \in T} x_{ij}^{kt} \quad (1)$$

subject to

$$I_0^t = I_0^{t-1} - \sum_{i \in V'} \sum_{k \in K} q_i^{kt} + d_i^t \qquad \forall t \in T \quad (2)$$

$$I_i^t = -d_i^t + I_i^{t-1} + \sum_{k} q_i^{kt} \qquad \forall i \in V', t \in T \quad (3)$$

$$I_i^t \leq L_i \qquad \forall i \in V', t \in T \quad (4)$$

$$\sum_{k \in K} q_i^{kt} \leq L_i - I_i^{t-1} \qquad \forall i \in V', t \in T \quad (5)$$

$$\sum_{i \in C} q_i^{kt} \leq Q_k \qquad \forall t \in T, k \in K \quad (6)$$

$$q_i^{kt} \leq y_i^{kt} L_i \qquad \forall i \in V', t \in T, k \in K \quad (7)$$

$$\sum_{j \in V'} x_{ij}^{kt} = \sum_{j \in V'} x_{ji}^{kt} \qquad \forall i \in V', t \in T, k \in K \quad (8)$$

$$\sum_{j \in V'} x_{ji}^{kt} = y_i^{kt} \qquad \forall i \in V', t \in T, k \in K \quad (9)$$

$$\sum_{j \in V'} x_{0j}^{kt} \leq 1 \qquad \forall k \in K, t \in T \quad (10)$$

$$\sum_{k \in K} y_i^{kt} \leq 1 \qquad \forall i \in V', t \in T \quad (11)$$

$$w_i^{kt} - w_j^{kt} + L_i x_{ij}^{kt} \leq Q_k - q_i^{kt} \qquad \forall i, j \in V', k \in K, t \in T \quad (12)$$

$$q_i^{kt} \leq w_i^{kt} \leq Q_k \qquad \forall i \in V', t \in T, k \in K \quad (13)$$

$$q_i^{kt} \geq 0, I_i^t \geq 0 \qquad \forall i, j \in V', k \in K, t \in T \quad (14)$$

$$x_{ij}^{kt} \; e \; y_i^{kt} \in \{0; 1\} \qquad \forall i, j \in V', k \in K, t \in T \quad (15)$$

The Function (1) aims to minimize DC's storage cost, costumer's storage cost and transportation's total cost. The equations (2) and (3) make a balancing of DC's and Costumers' storages, ensuring the attendance of the demand i in the period t. The inequations (4) and (5) secure that the quantity delivered to the costumer i in the period t will not overcome its available capacity. The inequation (6) ensure the total transported by the vehicle k will not exceed its own capacity. The inequation (7) decide which costumers will be visited in the period t. The equations (8) and (9) ensure the continuity of the vehicles' flow. The restriction (10) certifies all the vehicles to leave from DC. In (11), it has ensured all costumers for being visited by only one vehicle in t. The restrictions (12) and (13) ensure the non-competition for subtour. The inequation (14) secures the non-negativity of the variables in storages and transportation quantity decision. Finally, the restriction (15) forces the route decision and visiting variables and not the costumers to be binary ones

III. HEURISTIC USED IN THIS RESEARCH

There are two subproblems settled in IRP's, the vehicles routing and the storage management; the routing one will be discussed by the searches in a set of possible solutions. The local search consists of finding new neighbor solutions aiming to improve the value of the objective function. These neighbor solutions are generated by movements, each kind of them is considered a neighbor structure. Beside this, we also have the neighborhood search, which consists in making a several movements with the same local search's goal (to improve the objective function). In Routing Multi-Vehicle Problems, there are inter-route neighborhoods (movements among different routes) and intra-route ones (movements into the same route). The Table 4 contain some indexes do not presented before but used in the description of the investigated neighborhoods' structures.

Table IV: Neighborhood structures' indexes.

INDEXES	DESCRIPTION
Pos	Position of the vehicle k in the period t
K_1	Vehicle associated with t_1 and Pos_1
K_2	Vehicle associated with t_2 and Pos_2
t_1	Period associated with K_1 and Pos_1
t_2	Period associated with K_2 and Pos_2
Pos_1	Position associated with K_1 and t_1
Pos_2	Position associated with K_2 and t_2
A_{ij}	First arc $Arco_{ij}$ of the vehicle k in the period to be removed
$A_{i'j'}$	Second arc $Arco_{i'j'}$ of the vehicle k in the period to be removed

- INSERT (i, Pos, k, t): The neighborhood Insert consists in a inserting movement of a costumer i, in a specific position Pos, in the route of the vehicle k, in the period t.
- RANDOM INSERT (i, Pos, k, t): Similar to Insert, Random Insert consists in a random inserting movement of a costumer i, in a specific position Pos, in the route of the vehicle k, in the period t. The Figure 1 illustrates an example of "Insert" or "Random Insert".

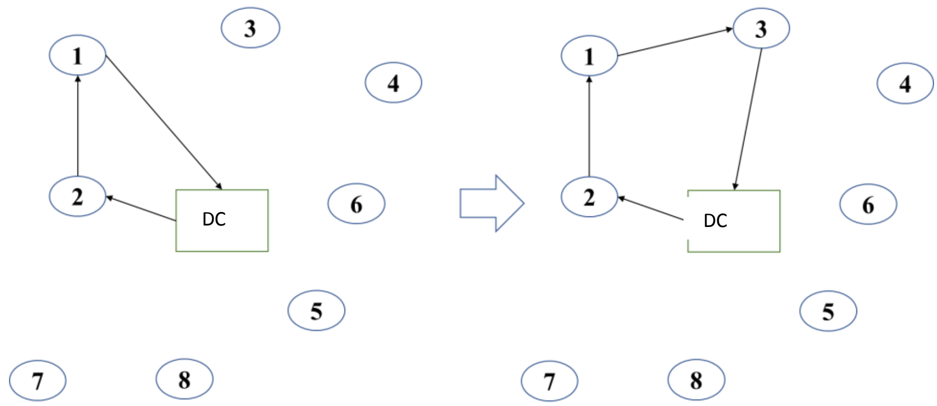

Fig.1. – Example of the neighborhood structure of Insert or Random Insert. Source: Authors (2021).

- REMOVE (Pos, k, t): This neighborhood is just a reverse movement of Insert, so that instead of inserting it removes costumers from the route, considering the position Pos of the route of the vehicles k in the period t. The Figure 2 illustrates the example of "Remove".

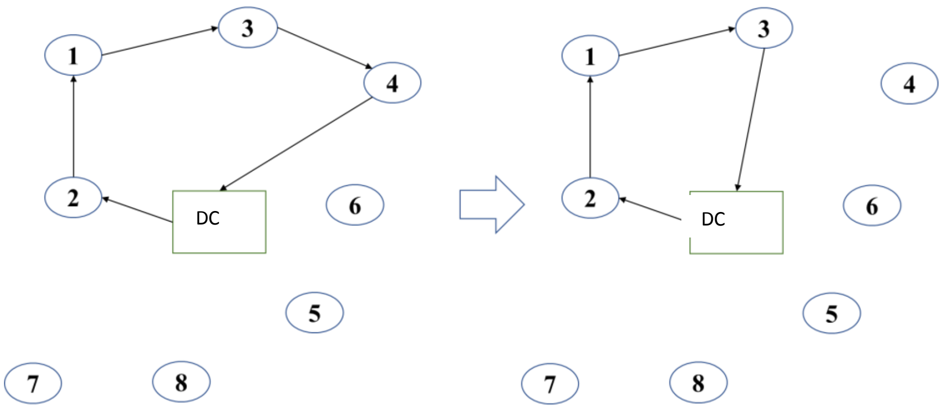

Fig.2. – Example of the neighborhood structure of Remove. Source: Authors (2021).

- SHIFT (Pos, k, t): This Neighborhood is intra-route and its movement is changing the position of two costumers always changing the position from the position Pos to the position Pos+1, route of the vehicle k in the period t. The whole neighborhood could be tested. The Figure 3 illustrates the example of "Shift".

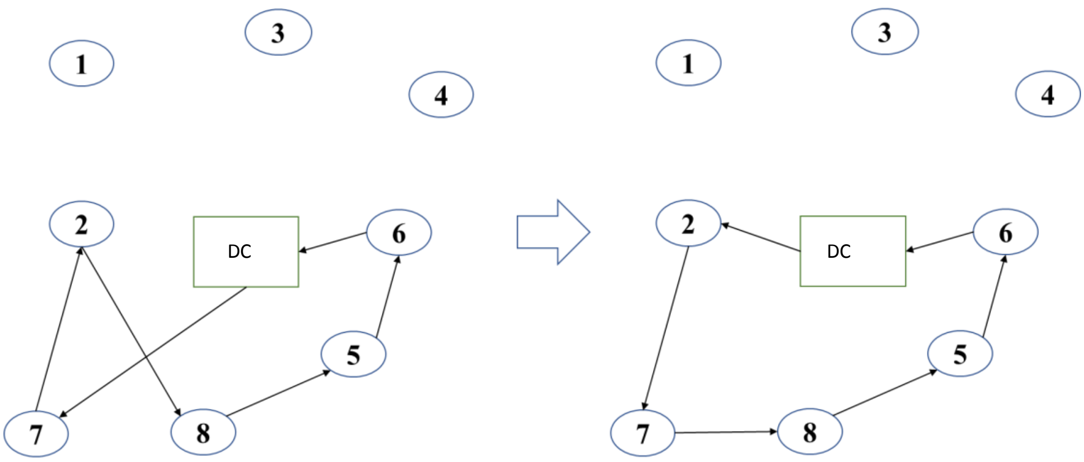

Fig.3. – Example of "Shift" neighborhood. Source: Authors (2021).

- SWAP $(Pos_1,\ k_1,\ t_1, Pos_2,\ k_2,\ t_2)$: In this neighborhood the movements are done by changing the positions among the costumers, from the costumer in Pos_1 in the route of k_1 in the period t_1 to the costumer in Pos_2 in the route of k_2 in the period t_2. This neighborhood was investigated completely. The Figure 4 contains an example of "Swap".

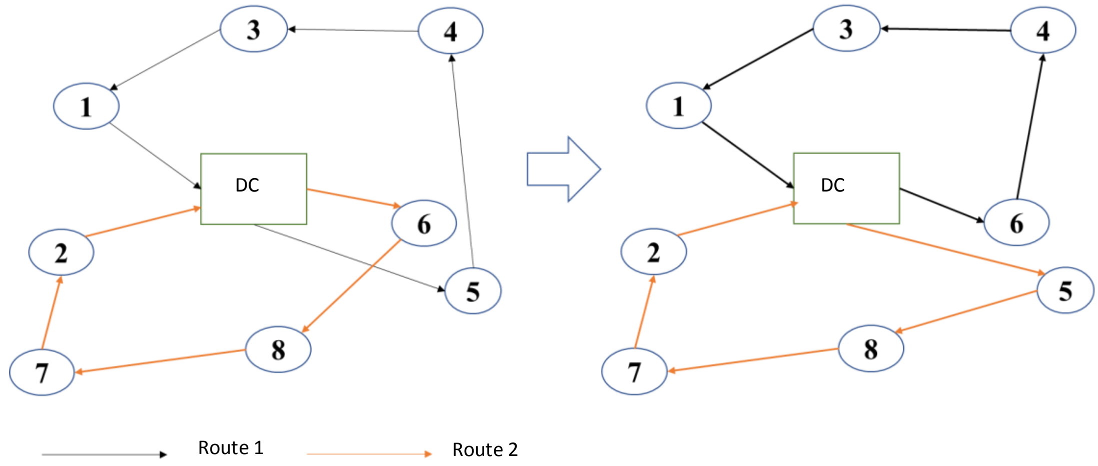

Fig.4. – Example of the Swap Neighborhood Structure. Source: Authors (2021).

- RELOCATE $(Pos_1,\ k_1,\ t_{1,}, Pos_2,\ k_2,\ t_{2,})$: This neighborhood structure removes a costumer from the position Pos_1 in the route k_1 in the period t_1 and inserts it to the position $Pos_2 + 1$, in the route k_2 in the period t_2. The neighborhood can be whole coursed as only one. The Figure 5 illustrates an example of "Relocate".

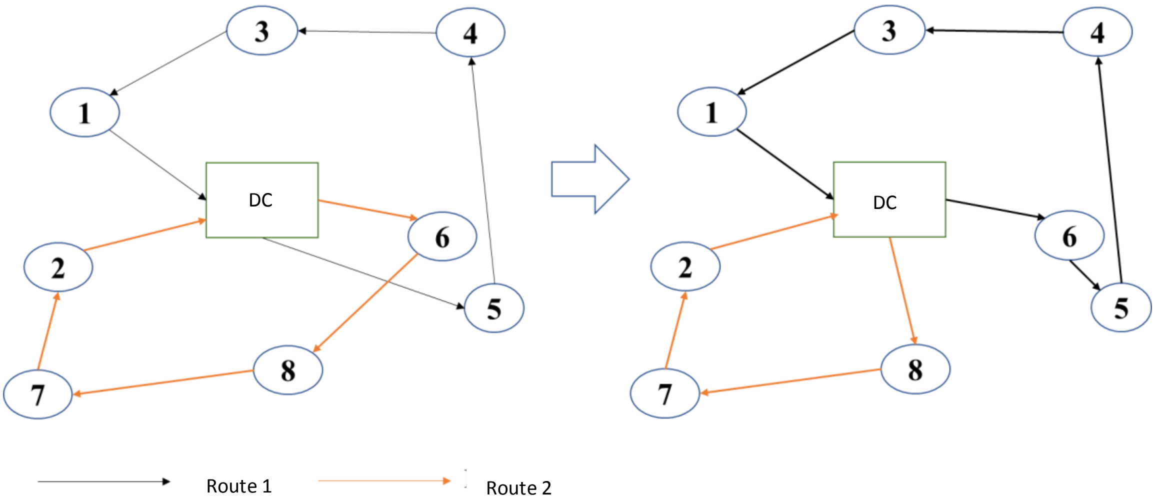

Fig.5. – Example of the Relocate Neighborhood Structure. Source: Authors (2021).

- 2OPT $(A_{ij}, A_{i'j'}, k, t)$: This is an intra-route neighborhood and its movement selects two arcs of the route k of the period t to be removed and inserts two new ones in their places; changing the position of the costumers involved in the process. In other words, selecting the arc (i, j) and the arc (i′, j′) to be replaced by the new arcs (i, i′) e (j, j′). The Figure 6 illustrates an example of the "2opt".

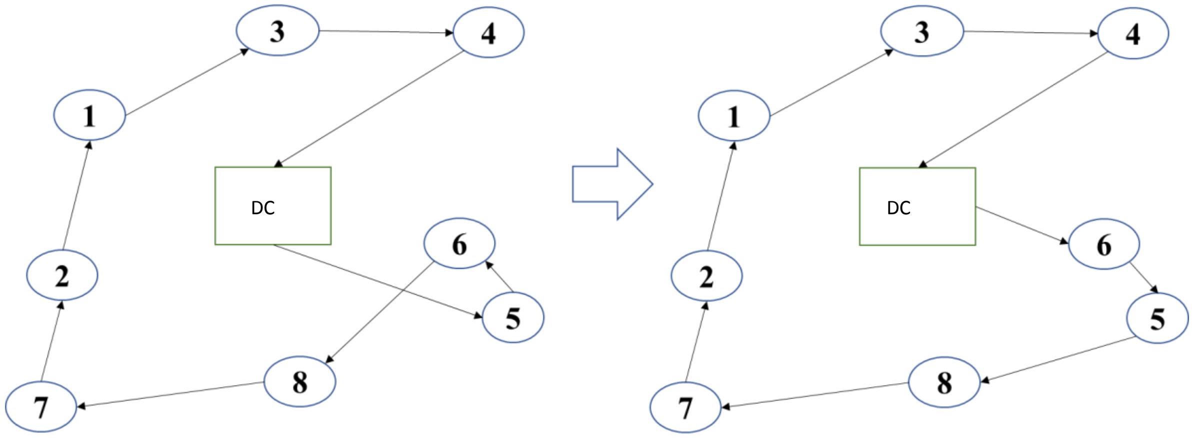

Fig.6. – Example of the "2opt" Neighborhood Structure. Source: Authors (2021).

Modeling the Minimum-Cost Maximum-Flow

The subproblem of Storage management was studied as a Minimum-Cost Maximum-Flow problem along the periods. This problem consists in defining the greatest possible value that can be sent from a node to another in a network with the lowest possible price. By this way, considering the Graph G (N, E) the application of the IRP problem and a graph G (N, E), In which there are two groups of edges, the intermediate edges are represented by the variable y_i^{kt} (visiting costumer decision). The edges and its costs and storage costs are considered with any or not any edges (same costumer's storage from a period to another one). There are also two groups of vertexes, in this case, the intermediate ones would be the vehicles k used in the specific period's route and its capacity. The other group of vertexes are the costumer, who have the ability to receive flow and demand. The problem presents the following characteristics:

- Maximum Capacity Flow either in arcs or in nodes (vehicles and costumers);
- Minimum Demand Product in the graph's nodes (costumers);
- Cost per unit of flow sent in the arc from each node to another (storages from a period to another);

The algorithm used to solve this problem was the Network Simplex, its mathematical definition, already adapted to IRP, is described bellow:

$$Max \sum_{I \in V} \sum_{k \in k} \sum_{t \in T} q_i^{kt} - \sum_{I \in V} \sum_{t \in T} I_i^t \qquad (16)$$

S.A.

$$I_0^t = I_0^{t-1} + d_i^t - \sum_{i \in V} \sum_{k \in K} q_i^{kt} \qquad \forall t \in T \qquad (17)$$

$$I_i^t = I_i^{t-1} + \sum_{k \in K} q_i^{kt} - d_i^t \qquad \forall i \in V, t \in T \qquad (18)$$

$$I_i^t \leq L_i \qquad \forall i \in V, t \in T \qquad (19)$$

$$\sum_{k \in K} q_i^{kt} \leq L_i - I_i^{t-1} \qquad \forall i \in V, t \in T \qquad (20)$$

$$q_i^{kt} \leq L_i y_i^{kt} \qquad \forall i \in V, k \in K, t \in T \qquad (21)$$

$$\sum_{i \in v} q_i^{kt} \leq y_i^{kt} Q_k \qquad \forall k \in K, t \in T \qquad (22)$$

$$I_i^t, q_i^{kt}, y_i^{kt}, q_i^{kt} \geq 0 \qquad \forall i \in V, j \in V \; j \neq i \; t \in T \qquad (23)$$

Regarding the original problem, these changes are the Objective Function, which aims to minimize for maximize and portion sizes changing, in the script restrictions and vehicles' flow conservation that cease to exist. The Function (16) aims to maximize the flow for the costumers (vertexes) minimizing the storages cost that were the edges ones, which is the edge of the costumer i in the period t for the same costumer i in the period t +1. The restrictions (17), (18), (19), (20), (21), (22), (23) have the same functions as the original problem's restrictions, being, respectively, (2), (3), (4), (5), (6), (7), (8). The Figure 7 illustrates an example the Minimum-Cost Maximum-Flow illustration for the IRP.

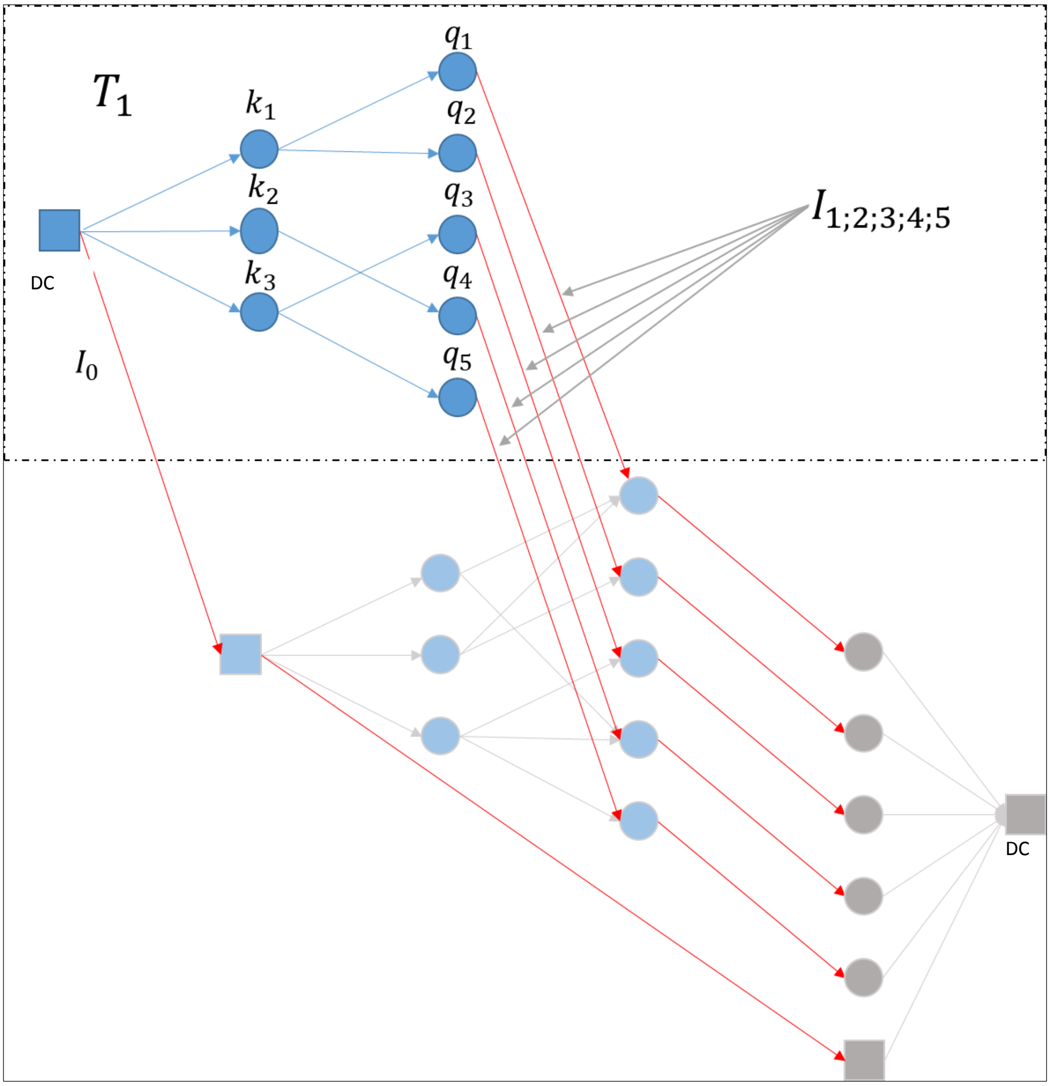

Fig.7. – Minimum-Cost Maximum-Flow illustration for the IRP. Source: Authors (2021).

Random Variable Neighborhood Descent and Simulated Annealing

Mladenovic e Hansen (1997) in their studies proposed an algorithm named Variable Neighborhood Descent (VND), it consists in doing the neighborhood sequence (inter-route and intra-soute). Inspired in VND, the researcher Subramanian (2012) proposed the VND variation named Random Variable Neighborhood Descent (RVND). The proposed version for Subramanian (2012) uses a random number to order the neighborhoods for being ran. In the case of the chosen random neighborhood being successful, the local searches have performed in the changed routes, the neighborhood is removed from the routes local list otherwise.

Let $N = \{N^1, N^2, N^3 \ldots N^z\}$ a set of neighborhoods structure, when a neighborhood structure does not succeed in improving the solution, the algorithm randomly chooses a new neighbor structure, which belongs to the same set and successively. The Table 5 contains the algorithm.

Table V: Algorithm RVND. Source: Adapted from Silva Junior (2013).

	Algorithm RVND
	Input: $f(.), N(.), r, s$
	Output: s
1	**Start**
2	Let r the number of neighborhoods;
3	Let L_r the neighborhood structure list;
4	L_r ← shuffle (L_r)

K ← + 1;
While ≤ r **do**
p ← $L_r(k)$;
Search for the best neighbor $s' \in N^{(p)}(s)$;
If $(f(s') < f(s))$ **then**
s ← s'
K ← 1;
Otherwise
K ← K ← + 1;
End
End
End
Return s;
End

The Simulated Annealing used in this research aims to improve the obtained solutions by RNVD and trying to leave optimum local for finding optimum global ones, towards this propose the SA accept random movements according to the algorithm's temperature. The efficiency of this metaheuristic is associated to the random movements accepted by the algorithm, considering the acceptance criterions and the local's search procedure.

Table VI: Algorithm RVND. Source: Adapted from Silva Junior (2013).

ALGORITHM SIMULATED ANNEALING
Input: Cooling Parameters
Output: Inviability
Start
S_0 ← ∞ +;
S ← RVND search the routing neighbors + NS;
While T > T_{min} **do**;
While equilibrium condition do:
Choose random solution $s' \in N(S)$
If s' *better than s* **then**
s ← s' accepts the solution
s ← RVND searches in routing neighborhoods + NS;
Otherwise
Accept s' with the probability $e^{\frac{-\Delta E}{T}}$
End
End-while
End-while
Return S

IV. COMPUTATIONAL RESULTS

To all the tests it was used the Julia programming language to codify the algorithm and the Atom (computer). The tests ran in a computer with the following characteristics: Intel® Core™ 2 i5, 2.2 GHz, 8 GB of main memory, and its operational system was Windows 10 64-bit.

To the initial tests and the validation of the algorithm it was applied a test to the models with existing instances on the literature available in the work of Coelho et al. (2012a) aiming to investigate the consistence and the algorithm's performance (available instances in: http://www.leandro-coelho.com/instances). By the end, it was compared the generated results between the exact algorithms of Coelho et al. (2012a) and the heuristic algorithm of Archetti et al. (2012). It was applied the algorithm to some great instances, in which the exact algorithm existed in the literature does not have a good performance. It was tested 24 instances being all of them with only 1 vehicle 10 with 50 to 100 customers and 4 vehicles to 200 customers.

Table VII: Comparison between the Exact Algorithm in the literature and the presented Algorithm.

INSTANCE	COSTUMER	EXACT SOLUTION	LOWER BOUND	ALG. PROP.	SE-ALG. PROP	LB- ALG. PROP
abs1n50.dat	50	**30189.40**	30189.40	30,487.53	0.99%	0.99%
abs2n50.dat	50	**29790.00**	29790.00	30,042.30	0.85%	0.85%
abs3n50.dat	50	**29790.90**	29790.90	29,958.23	0.56%	0.56%
abs4n50.dat	50	**31518.30**	31518.30	31,905.32	1.23%	1.23%
abs5n50.dat	50	**29240.40**	29240.40	29,420.32	0.62%	0.62%
abs6n50.dat	50	**31903.10**	31903.10	32,001.26	0.31%	0.31%
abs7n50.dat	50	**29734.50**	29734.50	30,051.49	1.07%	1.07%
abs8n50.dat	50	**25954.20**	25954.20	26,424.74	1.81%	1.81%
abs9n50.dat	50	**30192.90**	30192.90	30,345.23	0.50%	0.50%
abs10n50.dat	50	**31338.20**	31338.20	31,389.63	0.16%	0.16%
abs1n100.dat	100	57459.20	**57212.70**	57,578.28	0.21%	0.64%
abs2n100.dat	100	53510.10	**53076.40**	53,836.45	0.61%	1.43%
abs3n100.dat	100	58505.10	**58183.40**	58,752.38	0.42%	0.98%
abs4n100.dat	100	51554.20	**51511.00**	52,233.66	1.32%	1.40%
abs5n100.dat	100	57976.50	**57867.70**	58,620.51	1.11%	1.30%
abs6n100.dat	100	55087.80	**54843.00**	55,490.16	0.73%	1.18%
abs7n100.dat	100	56076.90	**55712.90**	56,294.44	0.39%	1.04%
abs8n100.dat	100	56057.10	**54729.00**	55,325.11	**-1.31%**	1.09%
abs9n100.dat	100	59425.90	**58086.80**	59,043.44	**-0.64%**	1.65%
abs10n100.dat	100	56588.30	**56034.30**	57,057.90	0.83%	1.83%
abs1n200.dat	200	136337.00	**109774.00**	111,588.65	**-18.15%**	1.65%
abs2n200.dat	200	141543.00	**111501.00**	112,818.42	**-20.29%**	1.18%
abs3n200.dat	200	123147.00	**106760.00**	108,208.07	**-12.13%**	1.36%
abs4n200.dat	200	129615.00	**107705.00**	109,422.99	**-15.58%**	1.60%

Table 7. Comparison between the Exact Algorithm in the literature and the presented Algorithm.

Table VIII: Comparison between the literature's heuristic results and the presented algorithm's results.

INSTANCE	COSTUMER	HAIR	ALG. PROP.	HAIR - ALG. PROP.
abs1n50.dat	50	30,225.36	30,487.53	0.87%
abs2n50.dat	50	29,856.26	30,042.30	0.62%
abs3n50.dat	50	29,904.15	29,958.23	0.18%
abs4n50.dat	50	31,677.87	31,905.32	0.72%
abs5n50.dat	50	29,400.33	29,420.32	0.07%
abs6n50.dat	50	31,946.33	32,001.26	0.17%
abs7n50.dat	50	29,768.03	30,051.49	0.95%
abs8n50.dat	50	26,521.96	26,424.74	**-0.37%**
abs9n50.dat	50	30,283.90	30,345.23	0.20%
abs10n50.dat	50	31,397.84	31,389.63	**-0.03%**
abs1n100.dat	100	57,721.23	57,578.28	**-0.25%**
abs2n100.dat	100	53,432.80	53,836.45	0.76%
abs3n100.dat	100	58,598.93	58,752.38	0.26%
abs4n100.dat	100	52,030.59	52,233.66	0.39%

abs5n100.dat	100	58,258.92	58,620.51	0.62%
abs6n100.dat	100	55,280.01	55,490.16	0.38%
abs7n100.dat	100	56,398.19	56,294.44	**-0.18%**
abs8n100.dat	100	55,384.47	55,325.11	**-0.11%**
abs9n100.dat	100	58,729.94	59,043.44	0.53%
abs10n100.dat	100	56,644.22	57,057.90	0.73%
abs1n200.dat	200	110,790.39	111,588.65	0.72%
abs2n200.dat	200	112,401.98	112,818.42	0.37%
abs3n200.dat	200	108,119.61	108,208.07	0.08%
abs4n200.dat	200	109,309.48	109,422.99	0.10%

Table 8. Comparison between the literature's heuristic results and the presented algorithm's results.

V. CONCLUSION

By this research, it has developed a hybrid heuristic method to solve an inventory routing problem. The method consists in a hybrid algorithm based in a neighborhood search for solving the transportation subproblem, to the other subproblem, the storage one, it was modeled as a Minimum-Cost Maximum-Flow and it was used the Network Simplex (NS) algorithm to solve it. For fleeing the optimum locals, it was applied the Simulated Annealing heuristic, which during the initial search phase uses a procedure RVND.

It was used seven neighborhood structures being six of them PRV and TSP ones. To evaluate the algorithm's performance, it has done computational tests with the literature's instances of 6 periods and from 50 to 200 costumers.

The results presented the efficiency of the method developed, once that, by just a few experiments, the method was six times better than the best literature's result and 5 times greater than the heuristic result. Future works may include the incorporation of more neighborhood structures aiming to improve the algorithm's efficiency. Adapt the heuristic to some variations of the IRP, such as PR, MMIRP, etc. Besides that, doing tests with another Minimum-Cost Maximum-Flow algorithms aiming to improve its efficiency.

REFERENCES

[1]. Adulyasak, Y., Cordeau, J-F, Jans R. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. INFORMS J. Comput. (26:1), pp.103-12, (2013).

[2]. Archetti, C., Bertazzi L., Hertz A., Speranza, M. G. A hybrid heuristic for an inventory routing problem. INFORMS J. Comput. (24:1), pp.101–116, (2012).

[3]. Archetti, C., Bertazzi, L., Laporte, G., Speranza, M. G. A branch and-cut algorithm for a vendor-managed inventory-routing problem. Transportation Sci. (41:3), pp.382–391, (2007).

[4]. Coelho, L. C., Laporte, G. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. Internat. J. Production Res. (1), pp.1-14, (2013a).

[5]. Coelho, L. C., Laporte, G. The exact solution of several classes of inventory-routing problems. Comput. Oper. Res. (40:2), pp.558–565, (2013b).

[6]. Coelho, L. C., Cordeau J-F., Laporte, G. Consistency in multivehicle inventory-routing. Transportation Res. Part C: Emerging Tech. (24:1), pp.270–287, (2012a).

[7]. Coelho, L. C., Cordeau, J-F., Laporte, G. The inventory routing problem with transshipment. Comput. Oper. Res. (39:11), pp.2537–2548, (2012b).

[8]. Dror, M., Ball, M. O., Golden, B. L. A computational comparison of algorithms for the inventory routing problem. Ann. Oper. Res. (4) pp.3–23, (1985).

[9]. Peres, I. T.; Repolho, H. M.; Martinelli, R.; Monteiro, N. J. Optimization in inventory-routing problem with planned transshipment: A case study in the retail industry. International Journal of Production Economics. (19), pp.748–756, (2017).

[10]. Subramanian, A. Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problem. Universidade Federal Fluminense, Niterói. . (2012)

Diego Moah, et. al. "Hybrid Metaheurístic to Inventory Routing Problem."*American Journal of Engineering Research (AJER)*, vol. 10(3), 2021, pp. 135-145.