Application of Hybrid Memetic Algorithm to solve Dynamic Vehicle Routing Problem with Overtime in the context of reverse logistics

Amina Berahhou[1] and Youssef Benadada[2]

***Abstract*— The Dynamic Vehicle Routing Problem (DVRP) is a complex and evolving challenge in transportation logistics, where constraints change over time. Traditional optimization methods struggle to adapt to such dynamic environments, necessitating more robust and flexible approaches. This paper explores the application of Memetic Algorithm (MA) to solve the Dynamic Vehicle Routing Problem with Simultaneous Delivery and Pickup, Overtime and Inventory Restriction (DVRPSDP-OT-IR) a special variant of DVRP. Experimental results demonstrate that the proposed approach significantly outperforms conventional heuristics and metaheuristics in terms of solution adaptability, computational efficiency, and cost minimization.**

I. INTRODUCTION

In recent years, road transport has made considerable progress in terms of performance, economic growth and the development of countries and companies. In parallel with this progress, the transport demand continues to grow over the years. This situation opens up opportunities and challenges for companies and manufacturers of transport systems. In real life, route optimization problems are very complex and represent a strategic issue for companies, because they are really exposed to dynamic environments. In a static environment, the search space, the objective function and the variables are all assumed to be constant in terms of time. This reduces the complexity of resolution, but it doesn't reflect the reality of things, because dynamism is the dominant character of real life. In fact, the inputs, procedures and outputs of a real system are all variable in terms of time.

The rapid evolution of the market, the increasing in customer requirements and the technological availability have forced researchers to reconsider their attitude. As a result, a new class of problem has emerged. These are dynamic combinatorial optimization problems. This class includes problems in which at least one of the components changes over time: the objective function, the variables, the constraints and the optimal solution. In this type of optimization, it is assumed from the very beginning that the information available is not complete, and that the solution initially provided is likely to change if new information becomes available. This is a multi-complicated challenge. From an algorithmic point of view, resolution algorithms must be able to update their data continuously and deliver optimal results in short time. From an executive point of view, dispatchers must have the technological and organizational means to collect new information and implement new solutions.

Another challenge facing transportation companies is the small fleet. Consequently, serving all customers during legal working hours is not always possible. As a result, vehicles have to use overtime. When the whole service cannot be performed during regular business hours, overtime is helpful. In other words, tolerating overtime enables managers to serve the highest number of customers before closing the working day, especially when demand is high. For the dynamic problem, you have to serve the maximum number of customers on the same day they make their requests. Overtime tolerance is certainly more beneficial in this case, especially if there are late requests. In cases where vehicles use overtime, this incurs significant cost for the business, especially when it comes to paying drivers and depot managers. In fact, this cost can be taken into account in two different ways: either by considering the total overtime, which reflects the overtime performed by all vehicles, or by considering the maximum overtime, which represents the overtime performed by the last vehicle returning to the depot. In both cases, we need to consider overtime as the objective to be minimized (Zargary and Samouei (2022))

On the other hand, a number important of countries have been orienting their strategies to the environmental sustainability of economic processes. Thus, the environmental protection has received attention and distribution companies start seeking for approaches that allow to integrate environmental considerations into supply chain management. A simple example of these approaches consists in optimizing the delivery of products from depot to customers and the simultaneous pickup of empty containers, waste or end-of-life products from customers to depot or recycling sites for repair, recycling or disposal at the minimum cost. In this context, researchers gathered those factors in the Vehicle Routing Problem with Simultaneous Delivery and Pickup (VRPSDP). It was introduced by Min (1989). In the VRPSDP, the customer has a delivery and pickup demands that must be performed simultaneously and each vehicle starts its tour from the depot and return to the depot at the end of the day to unload what has been collected. The inventory restriction is a necessary factor that requires more attention during all steps of the optimization. It is necessary to coordinate between the inventory and

*This work was not supported by any organization

[1]Amina Berahhou with the Smart Systems Laboratory, ENSIAS, Mohammed V University, Rabat, Morocco aminaberahhou@um5.ac.ma

[2]Youssef Benadada with the Smart Systems Laboratory, ENSIAS, Mohammed V University, Rabat, Morocco youssef.benadada@ensias.um5.ac.ma

the distribution and take into consideration the inventory capacity in delivery and pickup.

In this paper we treat an interesting variant of VRP that combine all those challenges which is the Dynamic Vehicle Routing Problem with Simultaneous Delivery and Pickup, Overtime and Inventory Restriction (DVRPSDP-OT-IR).

The remainder of the paper is organized as follows: Section II gives a description of the problem studied. Section III describes the steps and the processes of MA. In Section IV, the numerical results are presented. The conclusion is presented in Section V.

II. PROBLEM DESCRIPTION

The Dynamic Vehicle Routing Problem with Simultaneous Delivery and Pickup, Overtime and Inventory Restriction (DVRPSDP-OT-IR) is a special and complex variant of VRP, it combines the Dynamic version of the VRPSDP which is an NP hard problem and additional constraints such as inventory restrictions and the overtime.

This problem deals with the case of a transport company that aims to build routes for fleets of homogeneous vehicles, which leave from a single depot, serve a set of clients, and go back to the depot; each customer requests simultaneous delivery and pickup. This happens in a context where new orders arrive when the distribution plan has already been drawn up and the vehicles have started their tours, and we need to meet as many requests as possible on the same day. The inventory restrictions constraint means that the depot has two storage areas with limited capacity, the first is for the items that will be delivered to clients (DS: Stock for Deliveries), and the second is for the items collected from clients (PS: Stock for Pickups). Also, the vehicles can use the overtime if they cannot visit all customers during legal working time.

The problem studied is bi-objective. The first objective consists in minimizing the maximum overtime of all vehicles and the second objective minimizes the total cost of the routes. To treat the problem, the day is divided into time interval. The customers received after a time limit Tco (cut off time) will be served the next day. In the first time interval, the vehicles start their tours from depot and visit the clients remaining from the precedent day. For the other time intervals, the vehicles start their routes from the last customer visited in the previous time interval and serve the customers received during the precedent time interval and those that have not been visited by a vehicle (Montemanni et al. (2005)).

III. MEMETIC ALGORITHM

The memetic algorithm is a genetic algorithm (GA) that uses local search procedures to intensify the search. It was first named by Moscato (1989) to designate algorithms that hybridize genetic procedures such as crossover and mutation with local search procedures. A memetic algorithm (MA) is a combination of an Evolutionary Algorithm (EA) with local search. This combination has proved its effectiveness over traditional EAs in several optimization problems (Merz & Freisleben, 1999). Hart et al. (2005) have defined memetic algorithms (MA) as evolutionary algorithms that apply a local search process to refine the solutions of difficult problems. On their behalf, Zeng et al.(2012) developed an improved memetic algorithm to solve the VRP. The proposed algorithm uses a hybrid selection and crossover strategies. Regarding Ayadi & Benadada (2013), they used the memetic algorithm to solve a bi-objective VRP with multiple trips. They adopted a multiline chromosome representation. The MA was able to obtain high-quality results. On their part, Mańdziuk & Żychowski, (2016) solved the Dynamic VRP using the memetic algorithm. The algorithm was tested on a set of benchmarks from the literature and proved to be an effective method. Azad et al. (2017) treated the multi-depot and periodic VRP (MDPVRP) and developed a heuristic initialized stochastic memetic algorithm to resolve it. The algorithm proposed showed a high performance. For Ouaddi et al. (2020), they studied the multitours DVRP with overtime and proposed a memetic algorithm to solve it. The memetic algorithm proposed was able to outperform the ant colony system (ACS).On their part, Berahhou et al. (2022) have proposed an approach based on the memetic algorithm to solve the DVRPSDP. The different stages and mechanisms of the memetic algorithm are the following:

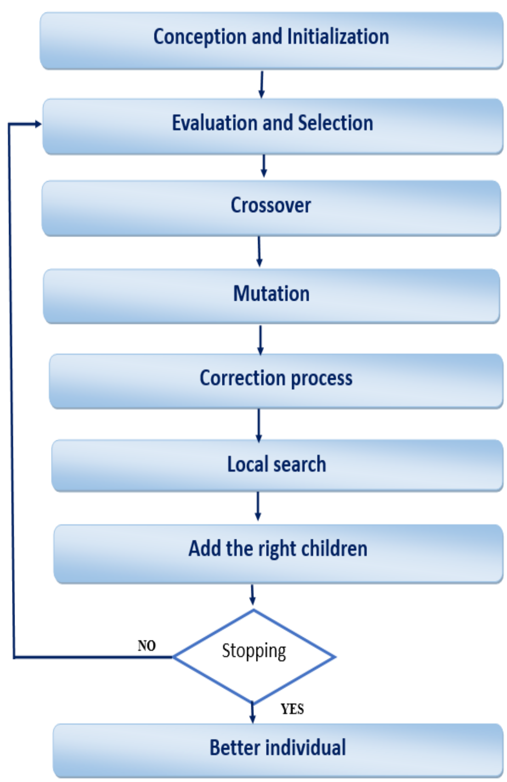

Fig. 1. The steps of Memetic Algorithm

A. *Coding the solution*

The population's chromosomes should effectively reflect all the essential information of the solution. We represent a chromosome by an array. Each chromosome represents the customers with the order in which they are served by a vehicle. Figure 2 shows the coding of a solution for 6 customers and 2 vehicles.

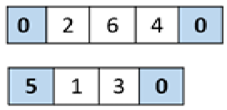

Fig. 2. Chromosome representation

B. *Initialization of the solution*

The memetic algorithm requires the creation of the initial population, which will be our starting point for the search of the optimal solution. We use the random insertion to generate feasible solutions that will serve as individuals in the initial population. The random insertion is a sequential insertion based on a random selection of customers. The algorithm of random insertion method is presented below (algorithm 1):

Algorithm 1 The algorithm of Random Insertion Method

```
1:  Choose a customer randomly
2:    Create a tour
3:      Choose a customer randomly
4:      If adding the customer to the tour violates the capacity constraint
5:        Go to 2
6:      If adding the customer to the tour violates the time constraint
7:        Go to 2
8:      If adding the customer to the tour violates the stock capacity constraint
9:        Go to 12
10:     Otherwise
11:       Add the customer to the tour
12:       If all customers are added
13: End
```

C. *Evaluation and selection*

The evaluation function represents the method used to assess the solutions; it compares individuals and determines the best one, in order to decide whether to insert it in the new generation or select the best solution at the end of the algorithm. The comparison is made as below: If one or both of the individuals to be compared surpasses the time horizon, the best individual is the one with the minimum overtime. If the time constraint is respected by both individuals, the better individual is the one with the lowest travel cost.

D. *Crossover*

The crossover is a genetic operator that combines two chromosomes called parents to produce a new one called child. We applied the uniform crossover which consists in generating randomly a binary mask of the same length as the parents. The (0) of the binary mask represent the positions preserved in P1, and the (1) of the binary mask represent the positions preserved in P2. To obtain C1, the elements not preserved from P1 are added with the same order that they have in P2. The same thing is done for C2 (Syswerda (1989)). Figure 4 shows an example of uniform crossover.

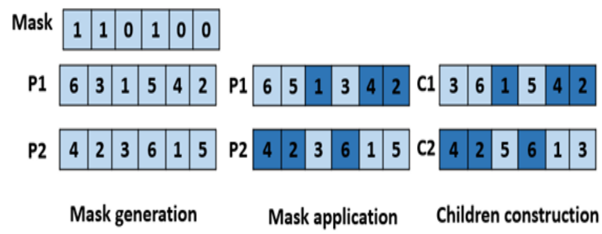

Fig. 3. Crossover Operation

E. *Mutation*

The mutation is a genetic operator that facilities the exploration of the research space, and prevents the algorithm from converging too quickly towards a local optimum. We adopted the random insertion method which consists in randomly selecting two tours. Then, select a customer from a tour and insert it into another tour at a randomly chosen location. Figure 5 shows an example of mutation operation.

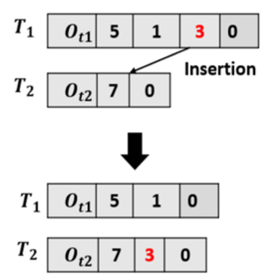

Fig. 4. Mutation Operation

F. *Local Search*

Local search is a heuristic optimization algorithm used when finding a global optimum is challenging due to a large search space. It focuses on refining a single candidate solution by iteratively moving to neighboring solutions. We applied two types of local search: local search intra-tour and local search inter-tour.

1) Local Search Intra-tour: The local search intra-tour refers to the exploration of solutions within the context of a single tour. Its goal is to improve the efficiency and the quality of individual routes. It consists in removing a customer from its current position in the tour and reinserts it at a different position.

2) Local Search Inter-tour: The local search inter-tour refers to the exploration of solutions by applying the swap and reintegration operations between two different tours of the solution. The swap consists in exchanging two customer visits between different tours. The reintegration consists in adding a customer visit from one tour to another at a specific position.

IV. Results

A. Benchmarks

To test the MA, we proposed test benchmarks that combine the dynamic character of the benchmarks of Kilby et al. (1998) with the fleet and time horizon constraints present in the benchmarks of Taillard et al. (1996). We used six instances of Christofides et al. (1979) (C) and twelve of Rochat and Taillard (1995) (Tai). To generate the delivery and pickup demands of each client from their original demand, we adopted the method used by Salhi and Nagy(1999).

For each instance, we proposed many values of N (Number of vehicles) and a bound for time interval T=[1,1*Z/N] with Z is the better result founded by Rochat and Taillard for the VRP. The arrival time of new customers is equivalent to the arrival time set by Kilby et al. (1998). The maximum permitted overtime is fixed to one quarter of the time horizon.

B. Results for small instances

In this section, we present the results of Cplex Solver and MA on small instances; we used the instances of Christofides et al.(1979). We utilized the instance C50X to generate 3 instances of 25, 30, and 35 clients.

Table I represents the results found by Cplex Solver and MA. The best solution is the one that obtain the shortest distance if the overtime is 0. Otherwise, the best solution is the one that minimizes the maximal overtime. We can notice that the MA was capable to give solutions very close to those obtained by CPLEX for all instances tested.

On the other hand, the execution time of Cplex is not comparable with the execution time of memetic algorithm (Table II). This was expected, since Cplex is founded on resolving mathematical model using exact methods. The execution time of MA is very short and it was capable to obtain results very comparable with Cplex.

TABLE I
Comparison of MA with Cplex solver

				Cplex Solver		Memetic Algorithm	
Inst	**m**	**T**	**Q**	**Distance**	**Overtime**	**Distance**	**Overtime**
C25X	1	577	80	514.21	0	517.62	0
	2	289	80	495.32	0	521.62	0
	3	192	80	508.21	0	498.35	0
	4	144	80	536.84	15.10	538.09	2.14
	5	115	80	529.08	22.74	539.94	9.36
C30X	1	577	80	543.61	0	539.82	0
	2	289	80	555.27	0	579.20	5.96
	3	192	80	536.74	0	569.13	0
	4	144	80	572.31	9.12	531.02	8.98
	5	115	80	541.87	4.39	582.02	14.98
C35X	1	577	160	502.31	0	557.50	0
	2	289	160	489.23	0	546.74	0
	3	192	160	554.23	2.78	543.25	0
	4	144	160	526.19	0	523.38	0
	5	115	160	517.25	0	596.36	20.8

TABLE II
Average of execution time in second

	Cplex	**MA**
Time (s)	115200	11

C. Results for large instances

Table III and Table IV represents the results obtained by solving DVRPSDP-OT-IR with memetic algorithm. As shown on the table MA was able to find feasible solutions for 144 of 158 instances. A feasible solution is a solution where the overtime does not surpass the maximal allowed overtime. The best solution is the solution with a minimum overtime if it surpasses 0. Otherwise, the best solution is the one with the minimal total distance. Concerning the coherence, the two objectives are coherent for all instances. i.e. for each instance, the best solution is the solution that gives the minimal distance and overtime (for the 4 executions done). Regarding the overtime obtained for all the instances, we notice that it is acceptable and does not exceed 200.

TABLE III
MA results for the DVRPSDP-OT-IR

			Memetic Algorithm					Memetic Algorithm	
Instance	**T**	**N**	**Distance**	**Overtime**	**Instance**	**T**	**N**	**Distance**	**Overtime**
c50X	577	1	608.31	8.5	c50Y	577	1	602.74	9.5
	289	2	645.85	13.6		289	2	653.24	23.9
	192	3	651.91	15.9		192	3	671.44	18.7
	144	4	668.54	42.8		144	4	675.82	25.6
c75X	919	1	972.14	0	c75Y	919	1	954.14	0
	459	2	987.05	14.3		459	2	974.28	11.8
	306	3	998.16	10.7		306	3	987.13	15.4
	230	4	1012.28	18.5		230	4	1014.30	20.4

TABLE IV
MA RESULTS FOR THE DVRPSDP-OT-IR

			Memetic Algorithm					Memetic Algorithm	
Instance	**T**	**N**	**Distance**	**Overtime**	**Instance**	**T**	**N**	**Distance**	**Overtime**
c100aX	909	1	1002.14	74.3	c100aY	909	1	988.27	84.3
	454	2	1027.61	50.7		454	2	997.25	46.8
	303	3	1036.80	31.2		303	3	1028.29	40.8
	227	4	1042.23	28.6		227	4	1021.31	39.2
c100bX	902	1	897.28	0	c100bY	902	1	905.14	0
	451	2	968.30	17.3		451	2	978.25	26.4
	301	3	917.26	8.4		301	3	935.44	23.6
	225	4	938.54	16.7		225	4	962.74	38.4
c150X	1131	1	1397.24	190.2	c150Y	1131	1	1412.69	194.2
	565	2	1347.29	74.3		565	2	1324.60	69.4
	377	3	1364.10	87.2		377	3	1374.15	95.7
	282	4	1407.12	40.2		282	4	1456.30	35.7
c199X	1421	1	1528.30	128.4	c199Y	1421	1	1539.32	137.2
	710	2	1546.21	86.1		710	2	1524.60	78.5
	473	3	1579.31	71.6		473	3	1548.12	63.2
	355	4	1587.18	68.4		355	4	1572.38	45.8
tai75aX	1780	1	1758.39	12.6	tai75aY	1780	1	1748.25	8.2
	890	2	1785.17	23.8		890	2	1806.54	19.2
	593	3	1778.29	26.5		593	3	1791.45	31.4
	445	4	1860.74	34.6		445	4	1882.31	48.5
tai75bX	1479	1	1379.30	0	tai75bY	1479	1	1345.41	0
	740	2	1396.37	0		740	2	1357.27	0
	493	3	1516.80	28.3		493	3	1394.84	34.91
	370	4	1564.25	40.7		370	4	1472.39	67.1
tai75cX	1420	1	1534.92	97.6	tai75cY	1420	1	1574.28	67.9
	710	2	1572.90	53.4		710	2	1617.29	80.4
	473	3	1586.92	65.7		473	3	1625.64	70.2
	355	4	1712.36	80.9		355	4	1839.2	102.5
tai75dX	1502	1	1476.38	0	tai75dY	1502	1	1486.3	0
	751	2	1518.36	35.7		751	2	1482.47	40.3
	501	3	1463.54	64.2		501	3	1479.70	76.8
	375	4	1547.29	91		375	4	1526.34	108.6
tai100aX	2245	1	2168.24	0	tai100aY	2245	1	2207.50	0
	1123	2	2167.28	50.3		1123	2	2236.71	75.0
	748	3	2184.11	87.6		748	3	2258.53	62.4
	561	4	2207.30	92.1		561	4	2271.28	108.3
tai100bX	2134	1	2052.34	0	tai100bY	2134	1	2184.44	0
	1067	2	2087.15	65.4		1067	2	2190.18	84.2
	711	3	2194.33	91.6		711	3	2279.50	96.8
	533	4	2175.28	113.2		533	4	2284.73	75.9
tai100cX	1547	1	1508	0	tai100cY	1547	1	1568.22	0
	773	2	1490.21	20		773	2	1517.88	0
	516	3	1513.10	60.2		516	3	1570.29	27.8
	387	4	1692.3	87.4		387	4	1687.25	49.3
tai100dX	1739	1	1764.24	29.7	tai100dY	1739	1	1813.52	34.2
	869	2	1752.41	18.7		869	2	1793.22	21.9
	580	3	1742.18	10.3		580	3	1804.21	16.7
	435	4	1767.29	28.4		435	4	1815.77	34.1
tai150aX	3361	1	3428.19	103.4	tai150aY	3361	1	3357.29	118.9
	1680	2	3398.26	42.9		1680	2	3347.11	68.2
	1120	3	3187.30	11.2		1120	3	3084.29	0
	840	4	3405.46	23.6		840	4	3287.22	21.9
tai150bX	3000	1	2938.18	0	tai150bY	3000	1	3007.28	5.2
	1500	2	3084.10	42.8		1500	2	3182.36	64.9
	1000	3	2947.43	26.4		1000	3	2996.10	45.3
	750	4	2910.14	37.5		750	4	2956.82	54.7
tai150cX	2595	1	2517.92	0	tai150cY	2595	1	2572.38	0
	1297	2	2524.36	12.4		1297	2	2521.94	15.2
	865	3	2673.55	39.4		865	3	2718.43	43.9
	649	4	2531.80	17.2		649	4	2574.22	28.4
tai150dX	2910	1	2941.30	10.5	tai150dX	2910	1	2953.89	8.4
	1455	2	3008.24	18.6		1455	2	3014.29	15.9
	970	3	2996.30	12.4		970	3	3005.67	9.6
	727	4	3015.78	21.8		727	4	3028.06	11.7

V. CONCLUSION

In this paper, we treated a new and special variant of the dynamic VRP, which combines simultaneous delivery and pickup with inventory restriction and the possibility of using the overtime by vehicles. The problem is interesting and can be the subject of several applications in the real world of transport. To solve it, we proposed a MA that combines genetic algorithm with a local search procedure. A series of numerical tests has been set up to test the effectiveness of the MA. The MA was able to get a feasible solution for all instances. In future works, we will solve the DVRPSDP-OT-IR with other metaheuristics in order to realize a comparative study.

REFERENCES

[1] S. Zargary and P. Samouei, Production-Routing-Inventory in Post-Disaster Conditions: a Multi-Objective Mathematical Model and Two Algorithms. Process Inte gration and Optimization for Sustainability, 6(4), 1163-1183, 2022

[2] H. Min, The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transportation Research, Vol. 23, pp 377–386, 1989.

[3] A. Berahhou and Y. Benadada, Dynamic vehicle routing problem with simultaneous delivery and pickup: Formulation and resolution, The 5th IEEE International Conference on Logistics Operations Management(GOL'20), DOI:10.1109/GOL49479.2020.9314759, 2020.

[4] P. Merz, and B. Freisleben, A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem, Proceedings of the 1999 Congresson Evolutionary Computation, CEC 1999, 3, 2063–2070

[5] R.Montemanni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati, Ant colony system for a dynamic vehicle routing problem, Journal of Combinatorial Optimiza tion, Vol. 10, pp. 327–343, 2005.

[6] P. Moscato, On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts:Towards Memetic Algorithms, Caltech Concurrent Computation Program (report 826), 1989

[7] J. Mańdziuk and A. Żychowski, A memetic approach to vehicle routing problem with dynamic requests, Applied Soft Computing Journal, 48, 522–534, 2016

[8] R. Ayadi and Y. Benadada, Memetic algorithm for a multi-objective vehicle routing problem with multiple trips, International Journal of Computer Science and Applications, Vol. 10, pp. 72–91, 2013

[9] W. E. Hart, N. Krasnogor, and J. E. Smith, Rcent Advances in Memetic Algorithms, In Studies in Fuzziness and Soft Computing (Vol. 166, Issue 1), 2005

[10] A. S. Azad, M. Islam, and S. Chakraborty, A heuristic initialized stochastic memetic algorithm for MDPVRP with interdependent depot operations, IEEE Trans actions on Cybernetics, 47(12), 430, 2017

[11] K. Ouaddi, F.-Z Mhada, and Y. Benadada, Memetic algorithm for multi-tours dynamic vehicle routing problem with overtime (MDVRPOT), International Journal of Industrial Engineering Computations 11 (2020) 643–662, 2020

[12] A. Berahhou, Y. Benadada, and K. Bouanane, Memetic algorithm for the dynamic vehicle routing problem with simultaneous delivery and pickup, International Journal of Industrial Engineering Computations, 13(4), 587-600, 2022

[13] Z. Zeng, Z. Xu, and W. Xu, An improved memetic algorithm for the vehicle routingproblem, 2012.

[14] G. Syswerda, Uniform Crossover in Genetic Algorithms, Proc, 3rd Int'l Conference on Genetic Algorithms, Morgan Kaufman Publishing, 1989.

[15] P. Kilby, P. Prosser, and P. Shaw, Dynamic VRPs: A Study of Scenarios, Report APES-06-1998, pp. 1–11, 1998.

[16] É. Taillard, P. Badeau, M.Gendreau, F. Guertin, and J.-Y. Potvin, (1997). A tabu search heuristic for vehicle routing problem wth soft time windows, Transportation Science,31 (2), 170–186, 1997.

[17] S. Salhi and G. Nagy, (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, Journal of the Operational Research Society, vol. 50, pp. 1034–1042, 1999.

[18] N. Christofides, The Vehicle Routing Problem, RAIRO Operations Research, Volume 10, pp. 55-70, 1976
[19] N. Christophides and J. Beasley, The period routing problem. Networks, Vol. 14, pp. 237-256, 1984.