

Dynamic Vehicle Routing Problem: Solution by Ant Colony Optimization with Hybrid Immigrant Schemes

Dhanya K.M.

Research Scholar, Dept. of Computer Science & Engineering, Pondicherry Engineering College,
Puducherry-605 014, India
E-mail: dhanyakm@pec.edu

Dr. S.Kanmani

Professor, Dept. of Information Technology, Pondicherry Engineering College, Puducherry-605 014, India.
E-mail: kanmani@pec.edu

Abstract—During past decades, several Meta-Heuristics were considered by researchers to solve Dynamic Vehicle Routing Problem. In this paper, Ant Colony Optimization integrated with Hybrid Immigrant Schemes methods are proposed for solving Dynamic Vehicle Routing Problem. Ant Colony Optimization with hybrid immigrant schemes methods namely HIACO-I, HIACO-II and HIACO-III focused on establishing the proper balance between intensification and diversification. The performance evaluation of the algorithms in which Random Immigrants and Elitism based Immigrants were hybridized in different proportions and added to Ant Colony Optimization algorithm showed that they had produced better results in many dynamic test cases generated from three Vehicle Routing Problem instances.

Index Terms—Meta-Heuristics, Dynamic Vehicle Routing Problem, Ant Colony Optimization, Hybrid Immigrant Schemes, HIACO-I, HIACO-II, HIACO-III, Intensification, Diversification, Random Immigrant, Elitism based Immigrant.

I. INTRODUCTION

Dynamic Vehicle Routing Problem (DVRP) is the most challenging Vehicle Routing Problem (VRP) in logistics and transportation fields. VRP is a combinatorial optimization problem which identifies the optimal paths the vehicles can perform to satisfy the demands of customers under certain conditions such as each customer specified earlier should be visited only once and the capacity of the vehicles should not exceed the capacity limit [3, 4, 5, 9, 13, 17, 23]. The VRP concept introduced by Dantzig and Ramser in 1959 can be represented as a directed graph $G(V, E)$, where $V = \{0, 1, 2, \dots, n\}$ denotes the set of customers except for the node $j=0$ which represents the depot and E the edge between two customers [15, 16, 18, 25, 28].

DVRP which is a variant of VRP and a dynamic optimization problem dates back to Speidel (1976) and

Psaraftis (1980) [7, 18]. One of the objectives of DVRP is to minimize the distance travelled by vehicles when they are routed to customers exposed to dynamic environments. The dynamic changes may be due to dynamic requests by customers, the traffic jam of vehicles and vehicles breakdown [11]. In dynamic optimization problems, the optimum during dynamic changes is to be tracked [6, 19, 24, 26]. Meta-Heuristic methods such as Variable Neighborhood Search, Genetic Algorithm, Artificial Bee Colony and Particle Swarm Optimization were intensively experimented in the past years to solve dynamic optimization problems [6, 7, 8, 12, 27, 29, 33].

Ant Colony Optimization (ACO) algorithm, a meta-heuristic method that converges to better quality solutions in Static VRP was also used to solve Dynamic VRP [1, 11]. ACO achieved the capability to deal with dynamism by incorporating immigrant schemes [24]. One of the immigrant schemes, random immigrant when incorporated with ACO leads to high level of diversity. The proposed work focused on the proper balance of intensification and diversification capabilities of ACO algorithms. The algorithms HIACO-I, HIACO-II and HIACO-III were introduced in which the random immigrants and elitism based immigrants were hybridized in different proportions and integrated with ACO algorithm to solve DVRP. To evaluate the performance of the proposed hybrid algorithms, they were compared with other ACO algorithms incorporating immigrant schemes on different dynamic environments constructed from three well-known VRP instances.

The paper is divided into five sections; Section I gives a brief Introduction. Section II reviews Ant Colony Optimization. Section III introduces the Proposed Work and is divided into subsections Ant Colony Optimization with Hybrid Immigrant Schemes and Performance Analysis. The Section IV reports on Results and Discussion of the Evaluation and Section V provides the Conclusion.

II. OVERVIEW OF ANT COLONY OPTIMIZATION

Ant Colony Optimization proposed by M.Dorigo is a Swarm intelligence based algorithm in which the behaviors of natural ants were considered in framing the algorithm [2, 14, 20, 30, 35]. In ACO algorithm, a population of ants (μ) is considered and each ant represents a solution of the problem to be solved [21, 22, 23, 24]. Ants start constructing the solutions from depot since vehicles are at the depot initially. Each ant chooses the next customer to be visited based on probability calculated in terms of pheromone trails (τ) and heuristic information. The probability of ant k , (p_{ij}^k) to visit the customer j from customer i is given by the Equation (1).

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{i \in N_i^k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta}, & \text{if } j \in N_i^k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where τ_{ij} represents the existing pheromone trail on the edge between customers i and j and η_{ij} represents the heuristic information available from distance between customers i and j (i.e., $1/d_{ij}$). N_i^k represents the unvisited customers in the neighborhood of customer i for ant k . α and β are the parameters that determine the relative influence of pheromone trail and heuristic information respectively. If the selection of the next customer exceeds the vehicle capacity, then the vehicle returns to the depot and a new vehicle should begin the route from the depot. This process continues until all the customers' requests are satisfied and thus an ant constructs a feasible VRP solution. When all the ants have constructed feasible solutions, it marks the completion of an iteration t and generation of population $P(t)$. A short term memory with size denoted by k_s is included in the pheromone matrix. The k_s best ants of $P(t)$ are added to short term memory at the end of iteration and modification of pheromone will be carried out. Ants follow the path rich in pheromone and converge to it. This is the stagnation property where the ants get trapped in confined regions and sometimes results in poor performance of the algorithm.

To avoid the stagnation property of ACO, various strategies such as local and global restart strategies, pheromone manipulation schemes, immigrant schemes, memory based methods and memetic algorithms have been adopted [22,23].The immigrant schemes such as random immigrant, elitism based immigrant and memory based immigrant were introduced to maintain diversity throughout the run. ACO with random immigrant (RIACO) performed too much diversification leading to diversification whereas ACO with elitism based immigrant (EIACO) and ACO with memory based immigrant (MIACO) generate guided diversity by transferring knowledge from previous environments [24].Afterward, Memetic algorithms (M-ACO) in which ACO algorithm hybridized with Local Search operator were introduced. Better intensification was achieved by

the method, but diversification needed for dynamic optimization problems was not attained. Hence, a method in which M-ACO hybridized with random immigrant was adopted to achieve a balance between diversification and intensification. But it is stated that in some dynamic cases on which the method was applied, the diversity scheme destroyed the improvement achieved with Local Search operator [23, 21]. Therefore, ACO integrated with hybrid immigrant schemes methods were proposed to maintain a good balance between intensification and diversification of ACO and to make it suitable for dynamic optimization problems.

III. PROPOSED WORK

The proposed work aimed at developing Ant Colony Optimization with hybrid immigrant schemes methods for dynamic vehicle routing problem. The performance of the proposed algorithms was evaluated on dynamic VRP instances constructed from three VRP instances. A comparative analysis with other ACO algorithms was also carried out.

A. Ant Colony Optimization with Hybrid Immigrant Schemes

The methods in which immigrant schemes were hybridized and combined with ACO in three different manners have experimented. The methods in which immigrant schemes like random immigrants and elitism based immigrants were hybridized in different proportions and incorporated to ACO were considered as ACO with hybrid immigrant schemes (HIACO) [32].

In the first method, denoted as HIACO-I, random immigrants and elitism based immigrants were used in equal proportions, i.e., 50% each with ACO. The random immigrants help HIACO-I to handle diversity and elitism based immigrants to use knowledge from the previous environment in dynamic environments. Then the method HIACO-II was proposed where random immigrants have used in more proportion 75% together with 25% elitism based immigrants in ACO. The elitism based immigrants were used to reduce the randomness of random immigrants in slightly changing environments. The third method, HIACO-III was introduced in which 75% of elitism based immigrants and 25% random immigrants were hybridized and added to ACO. The random immigrants in HIACO-III make elitism based immigrants address significantly changing environments also. The various ACO algorithms with immigrant schemes were classified and as shown in Table 1.

The random Immigrants introduced by Grefenestette were generated randomly to replace the worst ants in the population. The number of immigrants generated would be $r_i * k_s$, where r_i denoted the immigrant replacement rate. To generate Elitism based Immigrants introduced by Yang; the best ant from the previous environment was selected. The swaps between customers of the same route of the ant were performed with a mutation probability p_i^m . The $r_i * k_s$ elitism based immigrants generated from the previous environment $P(t-1)$ would replace the worst

ants in short-term memory [10, 24, 31, 33].

Table 1. Classification of ACO with Immigrant Schemes for DVRP

Ant Colony Optimization with Immigrant Schemes	Algorithm	Description
Ant Colony Optimization with Single Immigrant Scheme (SIACO)	RIACO	ACO with Random Immigrant
	EIACO	ACO with Elitism Based Immigrant
	MIACO	ACO with Memory Based Immigrant
Ant Colony Optimization with Hybrid Immigrant Scheme(HIACO)	HIACO-I	50% Random Immigrant & 50% Elitism Based Immigrant
	HIACO-II	75% Random Immigrant & 25% Elitism Based Immigrant
	HIACO-III	75% Elitism Based Immigrant & 25% Random Immigrant

B. Performance Analysis

The SIACO and HIACO algorithms were tested on dynamic test cases generated from VRP problem instances. The analysis focused on the effectiveness of HIACO algorithms to handle DVRP while establishing an equal importance to intensification and diversification factors. The best performance and diversity attained by an algorithm were mainly used to determine the effectiveness. The experiment design, performance measures and performance evaluation method considered are discussed below.

1) Experiment Design

The purpose of the experiments was to find out whether the proposed ACO algorithms maintains a good balance between intensification and diversification using hybrid immigrant schemes than other ACO algorithms with immigrant schemes on DVRP.

To carry out the experiments, Dynamic VRP instances were generated from three well known static VRP instances A-n32-k5, E-n76-k7 and F-n45-k4. The VRP instance, A-n32-k5 comes under "Class A" of Augerat et al instances. It is an instance with 32 customers and 5 minimum numbers of vehicles. The best value known for the instance is 784. E-n76-k7 belongs to "Class E" VRP instances proposed by Christofides and Eilon. E-n76-k7 is an instance with 76 customers and a requirement of 7 routes and its best value known is 683. F-n45-k4 is a "Class F" VRP instance proposed by Fisher. It contains 45 customers and 4 routes and its best value known is 724 [34].

The dynamic test cases were generated using Dynamic Benchmark Generator for Permutation Encoded Problem (DBGP) proposed by M.Mavrovouniotis. In DBGP, the dynamic changes were introduced by making modifications to the encoding of the static problem instance. For example, dynamic VRP test cases could be constructed by interchanging the location of a node with the location of another node in static VRP instance. The dynamic changes in DBGP depend on the magnitude of

change, the frequency of change and dynamic change environment [24]. The magnitude of change or change degree (m) is represented by a value that determines the number of swapped nodes. If the value of m is 0.25, it means that one fourth of the nodes can be replaced by other left out nodes in the problem instance. The frequency of change or change speed (f) denotes the number of times the nodes can be swapped. It depends on the algorithmic iterations. The dynamic change environment can be of random and cyclic change mode. The random dynamic environment is the one in which the reoccurrence of previously constructed dynamic environments in the current dynamic environment is not guaranteed. The latter one, cyclic dynamic environment can again be classified based on the way the base states of the dynamic environment are selected. If base state selection is in a logical ring fashion, it is called as reappear cyclically and referred as reappear randomly in random manner selection.

In the proposed work, DBGP parameters values were set as follows. The frequency of change was set as 10 and 100, the magnitude of change was given the values 0.25, 0.5 and 0.75 and the type of dynamic environments considered was Random (R), Reappear Random (RR) and Reappear Cyclic (RC). The number of base states used in cyclic dynamic environments was 4. So, the dynamic environment cycles 25 times when $f=10$ for 1000 iterations. For each instance, 18 dynamic environments were created.

The parameters of ACO algorithms with various immigrant schemes were initialized with values and those details are depicted in Table 2.

Table 2. Parameter Values used in SIACO and HIACO Algorithms for DVRP

Parameter	Value	Description
μ	50	Population size of ants
α	1	Parameter that determine the relative influence of pheromone trail (τ)
β	5	Parameter that determine the relative influence of heuristic information (η)
k_s	12	Size of Short term memory
r_i	0.4	Immigrant ant replacement rate
p_m^i	0.01	Immigrant ants' mutation probability

2) Performance Measures

Offline performance, total diversity and execution time are some of the metrics used to evaluate the HIACO and SIACO algorithms and are defined below [21, 22, 23, 24].

Offline Performance is a measure to evaluate the performance of the algorithm and is given by the Equation (2).

$$\overline{P_{OFF}} = \frac{1}{G} \sum_{i=1}^G \left[\frac{1}{R} \sum_{j=1}^R P_{ij}^* \right] \quad (2)$$

where G represents the total number of generations, R denotes the number of runs and P_{ij}^* represents the best fitness value after a change in generation i of run j .

Total Diversity measures the difference between the solutions in a population and is calculated using the Equation (3).

$$\overline{T_{DIV}} = \frac{1}{G} \sum_{i=1}^G \left[\frac{1}{R} \sum_{j=1}^R DIV_{ij} \right] \quad (3)$$

where G is the total number of iterations or generations, R is the number of runs and DIV_{ij} is the population diversity which is defined by the Equation (4).

$$DIV_{ij} = \frac{1}{\mu(\mu-1)} \sum_{p=1}^{\mu} \sum_{q \neq p}^{\mu} S(p, q) \quad (4)$$

where S (p, q) is the similarity metric .For VRP, S (p, q) is given by the Equation (5).

$$S(p, q) = 1 - \left(\frac{CE_{pq}}{n + \text{avg}(n_{V_p}, n_{V_q})} \right) \quad (5)$$

where CE_{pq} are the common edges of the customers between the ant p and ant q, n_{V_p} and n_{V_q} are the numbers of vehicles used in the ant p and ant q. If the value of S (p, q) is 0, it means that the two ants are similar. A range of values between 0.6 and 0.8 is considered to be good since it means that the ants can communicate with dissimilar ants in a good manner.

Execution Time is the time taken to execute the algorithm. It corresponds to the CPU time or wall clock time. But the drawback is that it depends on various computer characteristics like hardware, operating system, language and compilers on which the algorithm is executed.

3) Performance Evaluation

For performance assessment of proposed hybrid algorithms, they were compared with other ACO algorithms incorporating single immigrant scheme. The different dynamic test cases for evaluation were generated from each of the three VRP instances by varying the parameters such as change mode, change degree and change speed and it is shown in Table 3. In total, 324 experiments were conducted by SIACO and HIACO algorithms on 54 dynamic test cases.

In various dynamic environments, each algorithm was executed for 30 runs and 1000 iterations. The minimum distance obtained in each run and their corresponding iterations were noted. Then the smallest value from those minimum distance values was taken into account as the best solution. If that smallest distance value was obtained in more than one runs, then the run in which the smallest distance value was first noticed could be considered as the best run of that algorithm.

Depending on the best solution obtained by the algorithms, the algorithm that performed better was identified for each dynamic test case. The algorithm that produced the lowest value among the best solution of each algorithm was noted and considered as the algorithm that performed better than all other algorithms.

If different algorithms have produced the same lowest distance value on a particular dynamic test case, then the algorithm that performed better was decided on the basis of the best run of the algorithms. In case, the lowest distance value was obtained in the same run for different algorithms, then iteration in the particular run and execution time of the algorithm were also taken into account to determine the algorithm that performed better. For example, in a random dynamic environment created with $f=100$ and $m=0.5$, the best solution 813 was obtained by EIACO in iteration 531 of run 29 and by HIACO-III in iteration 557 of run 16. Then, HIACO-III was considered as the algorithm that performed better.

Table 3. Dynamic Test Cases generated from VRP Instances

Dynamic Test Cases	Dynamic Change Parameters		
	Change Mode	Change Degree	Change Speed
1	R	0.25	10
2	RR		
3	RC		
4	R	0.5	
5	RR		
6	RC		
7	R	0.75	
8	RR		
9	RC		
10	R	0.25	100
11	RR		
12	RC		
13	R	0.5	
14	RR		
15	RC		
16	R	0.75	
17	RR		
18	RC		

For all dynamic test cases, the diversity value obtained in the best run and iteration of the better performed algorithm was also recorded. The SIACO and HIACO algorithms that performed better on 18 dynamic test cases (TC) of each of the three instances-A-n32-k5, E-n76-k7 and F-n45-k4 and values of their performance parameters such as distance (Dist.), diversity (Div.) and execution time (Time) along with the run (R) and iteration (G) in which it was first obtained are shown in Tables 4-6.

Table 4. Performance of SIACO and HIACO Algorithms on A-n32-k5 generated Test Cases

TC	Algorithm	Dist.	R	G	Div.	Time (s)
1	HIACO-I	814	22	327	0.648175	53.296
2	HIACO-I	817	11	649	0.666304	53.639
3	MIACO	817	25	663	0.653957	52.733
4	EIACO	815	13	332	0.639512	52.811
5	HIACO-III	813	17	786	0.682982	53.859
6	RIACO	813	13	774	0.734014	53.999
7	MIACO	817	30	692	0.745295	53.124
8	HIACO-I	815	17	297	0.658866	54.264
9	MIACO	820	4	838	0.606406	52.983
10	EIACO	813	8	691	0.613345	51.062
11	MIACO	813	2	403	0.580964	50.828
12	EIACO	806	17	262	0.637925	51.077
13	HIACO-III	813	16	557	0.654172	51.733
14	HIACO-I	813	7	513	0.639592	51.937
15	HIACO-II	813	27	891	0.580454	52.218
16	HIACO-II	810	23	352	0.748515	51.953
17	HIACO-II	810	23	352	0.748515	51.655
18	HIACO-I	811	21	243	0.678662	51.343

Table 5. Performance of SIACO and HIACO Algorithms on E-n76-k7 generated Test Cases

TC	Algorithm	Dist.	R	G	Div.	Time (s)
1	EIACO	733	3	580	0.501573	130.639
2	HIACO-I	740	14	948	0.580289	131.687
3	HIACO-III	733	17	970	0.535127	129.312
4	HIACO-III	736	23	470	0.517402	130.702
5	EIACO	745	10	510	0.552982	131.406
6	HIACO-III	739	10	437	0.591508	132.874
7	HIACO-I	730	8	649	0.641453	132.734
8	RIACO	742	24	487	0.6922	129.843
9	MIACO	742	1	967	0.5099	128.39
10	HIACO-III	728	24	791	0.417302	109.436
11	HIACO-I	730	5	682	0.498497	112.39
12	HIACO-I	730	13	198	0.460617	111.437
13	HIACO-III	720	29	183	0.544988	111.64
14	HIACO-III	720	29	183	0.346227	113.39
15	EIACO	727	8	255	0.478308	111.031
16	HIACO-III	727	14	674	0.356083	110.39
17	MIACO	733	20	980	0.40443	107.796
18	HIACO-III	718	24	682	0.417855	110.936

Table 6. Performance of SIACO and HIACO Algorithms on F-n45-k4 generated Test Cases

TC	Algorithm	Dist.	R	G	Div.	Time (s)
1	HIACO-III	755	12	899	0.409643	66.906
2	HIACO-II	762	8	843	0.624673	67.39
3	HIACO-II	769	22	136	0.616421	66.905
4	HIACO-I	732	24	140	0.38344	68.781
5	RIACO	759	4	297	0.666858	69.421
6	HIACO-II	762	20	384	0.651743	69.874
7	MIACO	733	25	17	0.490774	67.593
8	MIACO	731	4	775	0.476018	68.296
9	RIACO	731	15	129	0.459267	68.874
10	HIACO-I	762	23	881	0.647598	63.405
11	HIACO-III	764	22	826	0.570476	61.858
12	HIACO-III	758	11	692	0.511862	62.53
13	HIACO-I	758	2	618	0.476626	64.139
14	HIACO-I	760	16	662	0.660422	64.812
15	HIACO-II	766	22	609	0.644	66.484
16	RIACO	730	17	162	0.51065	64.109
17	HIACO-I	730	2	932	0.399579	62.249
18	RIACO	730	7	222	0.516399	62.608

IV. RESULTS AND DISCUSSION

From the experiment results, it was observed that ACO with hybrid immigrant schemes performed better than ACO with single immigrant schemes in 63% of dynamic test cases and their better performance ratio is depicted in Figure1.

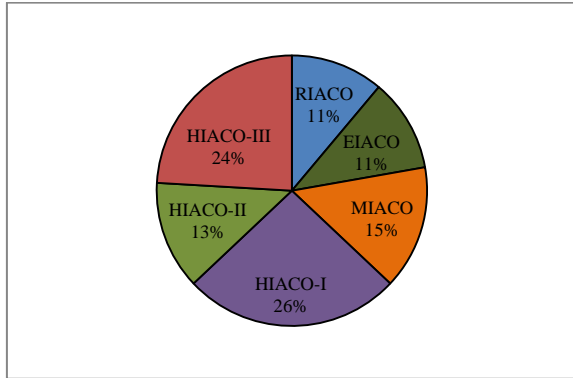


Fig.1. Performance Analysis of SIACO and HIACO Algorithms on DVRP Test Cases

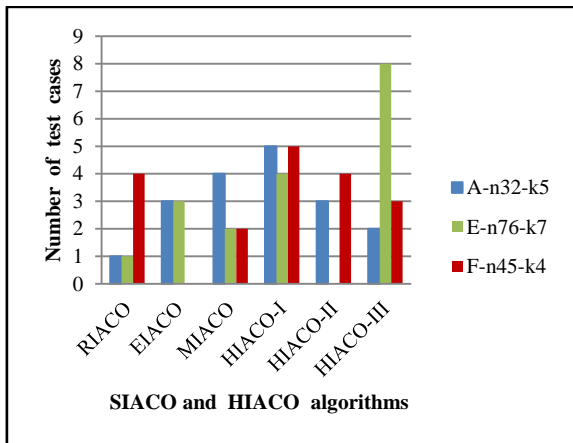


Fig.2. Performance Analysis of SIACO and HIACO Algorithms on Dynamic Test Cases with respect to VRP Instances

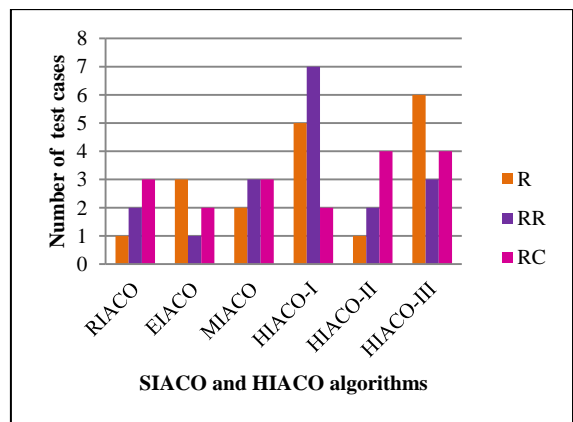


Fig.3. Performance Analysis of SIACO and HIACO Algorithms on Dynamic Test Cases with respect to Dynamic Environments

The number of dynamic test cases in which each SIACO and HIACO algorithms performed better considering VRP instances from which test cases were generated is shown in Figure 2 and based on type of dynamic environments is given in Figure 3.

HIACO-I performed better than all other algorithms on A-n32-k5 constructed dynamic test cases and its best performance was observed mainly in reappearing random dynamic environments. In the case of E-n76-k7 instance generated test cases, the outstanding performance was achieved by HIACO-III. The algorithm HIACO-III performed better mainly in random dynamic environments. HIACO-I also performed better than all other SIACO algorithms on E-n76-k7 dynamic test cases. Considering F-n45-k4 instance created test cases, HIACO-I performed better than all other algorithms. The better performance was obtained by HIACO-I in random dynamic environments on that instance.

It was observed that HIACO-III produced better results in random dynamic environments. HIACO-I performed better than all algorithms in reappear random dynamic environment irrespective of the instances. Based on the instance, HIACO-II and HIACO-III showed varying good results in the cyclic dynamic environment which reappears cyclically.

HIACO-I performed better than all algorithms on A-32-k5 instance generated dynamic environment with $m=0.25$ and $f=10$. In a dynamic environment with $m=0.25$ and $f=100$, HIACO-III performed significantly better than other algorithms on F-n45-k4 and E-n76-k7 generated instances. HIACO-III also performed better than all other algorithms on E-n76-k7 generated instance with $m=0.5$ for both values of f . For the dynamic environment with $m=0.75$ and $f=100$, A-n32-k5 generated instance produced better solutions by HIACO-II and HIACO-III produced better solutions on the instance created from E-n76-k7 in the same environment.

Considering the diversity factor, HIACO algorithms have obtained diversity values between 0.60 and 0.75 in almost all the dynamic test cases on A-n32-k5. But HIACO algorithms have attained diversity values in the good range only on a few dynamic test cases generated from other two instances. RIACO algorithms exhibited high diversity in all the dynamic test cases irrespective of the instances.

The execution time of algorithms increase as the size of the problem instances becomes big. Here, more execution time was taken by algorithms for E-n76-k7 dynamic test cases followed by F-n45-k4 and A-n32-k4 test cases. Similarly, more time were spent by algorithms for dynamic test cases with high frequency of dynamic change, i.e., $f=10$.

The intensification and diversification achieved by SIACO and HIACO algorithms were assessed in terms of lowest distance and corresponding diversity obtained in the particular run and iteration of the execution of the algorithm respectively. The intensification and diversification by SIACO and HIACO algorithms on various dynamic test cases generated from each of the three VRP instances are shown in Figures 4-9.

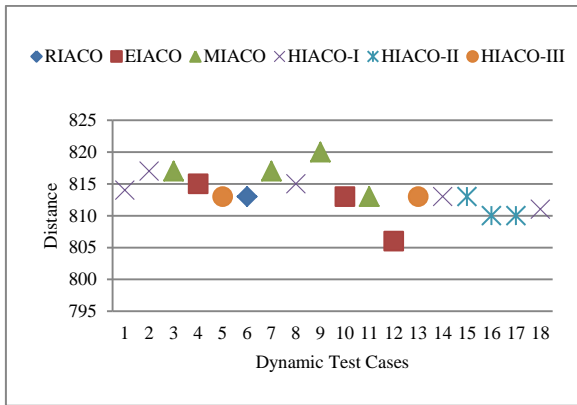


Fig.4. Performance Analysis of SIACO and HIACO Algorithms on A-n32-K5 generated Test Cases with respect to Intensification

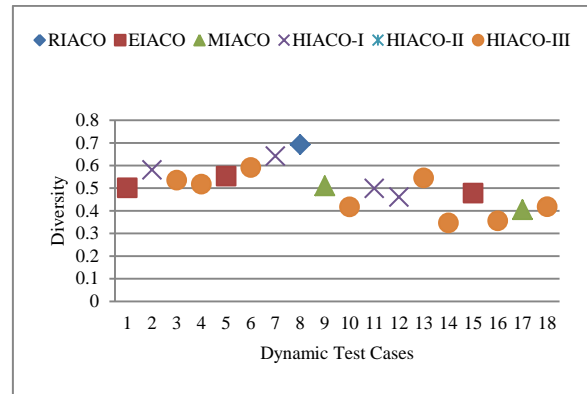


Fig.7. Performance Analysis of SIACO and HIACO Algorithms on E-n76-k7 generated Test Cases with respect to Diversification

The lowest distance values obtained by SIACO and HIACO algorithms on E-n76-k7 instance generated dynamic test cases lies in the range 720-745 and diversity values in the range 0.4-0.7 in almost all the cases. The algorithms were not able to achieve good diversification in E-n76-k7 instance except by HIACO-I and RIACO in one instance each. HIACO-III algorithms have shown a proper balance between intensification and diversification in a few cases only. HIACO-III algorithms with less diversity have better performance.

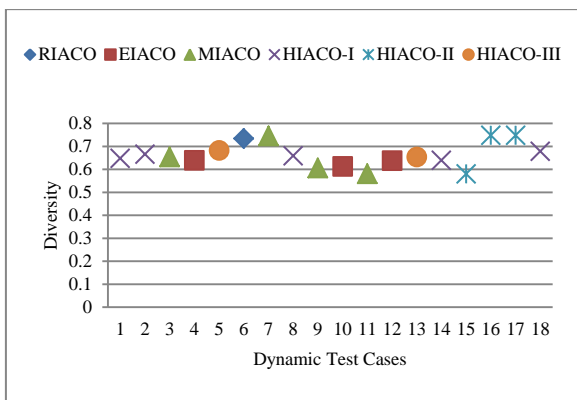


Fig.5. Performance Analysis of SIACO and HIACO Algorithms on A-n32-K5 generated Test Cases with respect to Diversification

In A-n32-k5 instance generated test cases, the lowest distance values obtained by algorithms were in the range 810-820 and diversity in the range 0.6-0.8 in almost all the cases. HIACO algorithms have maintained a good balance between intensification and diversification in most of the cases.

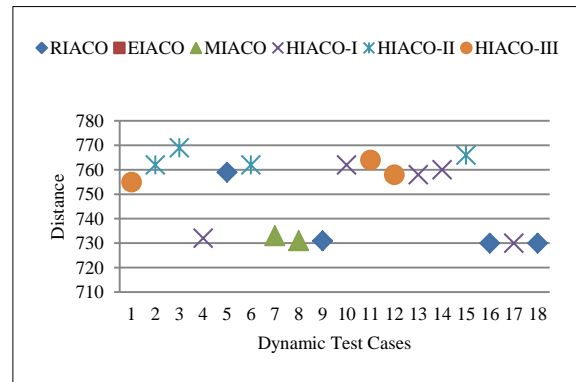


Fig.8. Performance Analysis of SIACO and HIACO Algorithms on F-n45-k4 generated Test Cases with respect to Intensification

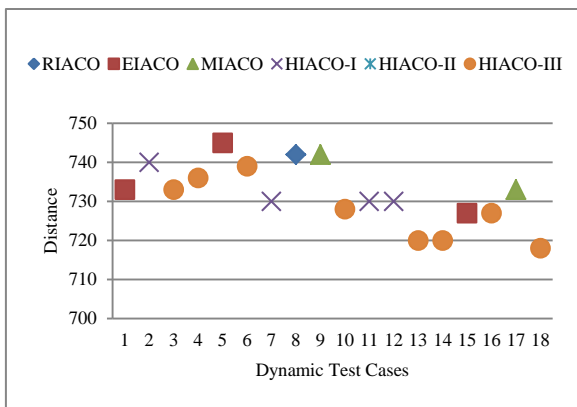


Fig.6. Performance Analysis of SIACO and HIACO Algorithms on E-n76-k7 generated Test Cases with respect to Intensification

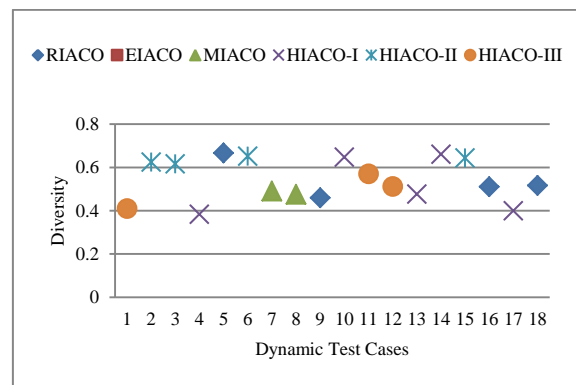


Fig.9. Performance Analysis of SIACO and HIACO Algorithms on F-n45-k4 generated Test Cases with respect to Diversification

In F-n45-k4 instance generated dynamic test cases, the lowest distance values obtained by SIACO and HIACO algorithms were in the range 730-770 and diversity values in the range 0.4-0.7 in almost all the cases. The algorithms which showed good diversity values have lower performance only.

V. CONCLUSION

The proposed Ant Colony Optimization with hybrid random immigrant and elitism based immigrant schemes algorithms such as HIACO-I, HIACO-II and HIACO-III had performed better than Ant Colony Optimization with single immigrant scheme algorithms in many Dynamic Vehicle Routing Problem test cases. HIACO-III outperformed all other algorithms on E-n76-k7 instance generated dynamic test cases. When the type of the dynamic environments are considered, HIACO-III performed better in random dynamic environments whereas HIACO-I in reappear random environments and both HIACO-II and HIACO-III have shown good performance in reappear cyclic environments.

HIACO algorithms have maintained a good balance between intensification and diversification on A-n32-k5 instance generated test cases. It can be concluded that the performance of the algorithms is dependent on the problem instance and type of the dynamic environment. As future research directions, the parameters of the algorithms can be varied and applied on various dynamic environments. Moreover, ACO or any other metaheuristic methods integrated with various hybrid immigrant schemes can be attempted.

REFERENCES

- [1] Barkaoui, M., Berger, J., Boukhtouta, A. Customer Satisfaction in dynamic vehicle routing problem with time windows. *Appl. Soft Comput.*, 2015, 35:423-432.
- [2] Boussaïd, I., Lepagnot, J., Siarry, P. A survey on optimization metaheuristics. *Inform. Sciences.* 2013, 237:82-117.
- [3] Brito, J., Martinez, F., J., Moreno, J., Verdegay, J., L. An ACO hybrid meta heuristic for close-open vehicle routing problem with time window and fuzzy constraints. *Appl. Soft Comput.*, 2015, 32:154-163.
- [4] Cattaruzza, D., Absi, N., Feillet, D., Vidal, T. A memetic algorithm for the multi trip vehicle routing problem. *Eur. J. Oper. Res.* 2014, 236: 833-848.
- [5] Chiang, T.,-C, Hsu, W.,-H. A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows. *Comput. Oper. Res.* 2014, 45:25-37.
- [6] Cruz, C., Gonzalez, J., R., Pelta, D., A. Optimization in dynamic environments: A survey on problems, methods and measures. *Soft Comput.* 2011, 15: 1427-1448
- [7] De Armas, J., Melián -Batista, B. Variable Neighborhood Search for a Dynamic Rich Vehicle Routing Problem with Time Windows. *Comput. Ind. Eng.* 2015, 85:120-131.
- [8] De Armas, J., Melián -Batista, B. Constrained dynamic vehicle routing problems with time windows. *Soft Comput.* 2015, 19:2481-2498.
- [9] De Oliveira, F., Enayatifar, R., Sadaei, H., J., Guimarães, F., G., Potvin, J.,-Y. A cooperative coevolutionary algorithm for the Multi-Depot Vehicle Routing Problem. *Expert Syst. Appl.* 2016, 43:117-130.
- [10] Eaton J., Yang, S., Mavrovouniotis, M. Ant colony optimization with immigrants schemes for the dynamic railway junction rescheduling problem with multiple delays. *Soft Comput.* 2016, 20: 2951-2966.
- [11] Euchi, J., Yassine, A., Chabchou, H. The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach. *Swarm Evol. Comput.* 2015, 21: 41-53.
- [12] Ghannadpour, S., F., Noori, S., Tavakkoli-Moghaddam, R. A multiobjective dynamic vehicle routing problem with fuzzy time windows: Model, Solution and Application. *Appl. Soft Comput.* 2014, 14: 504-527.
- [13] Jiang, J., Ng, K., M., Poh, K., L., Teo, K., M. Vehicle Routing Problem with a heterogeneous fleet and time windows. *Expert Syst. Appl.* 2014, 41:3748-3760.
- [14] Kar, A., K. Bio inspired computing – A review of algorithms and scope of applications. *Expert Syst. Appl.* 2016, 59:20-32.
- [15] Karakatic, S., Podgorelec, V. A survey of genetic algorithms for solving multi depot vehicle routing problem. *Appl. Soft Comput.* 2015, 27: 519-532.
- [16] Koç, C., Bektas, T., Jabali, O., Laporte, G. Thirty years of heterogeneous vehicle routing. *Eur. J. Oper. Res.* 2016, 249: 1-21.
- [17] Kourank Beheshti, A., Hejazi, S., R. A novel hybrid column generation-meta heuristic approach for the vehicle routing problem with general soft time window. *Inform. Sciences.* 2015, 316: 598-615.
- [18] Lin, C., Choy, K., L., Ho, G., T., S., Chung, S., H., Lam, H., Survey of Green Vehicle Routing Problem: Past and Future Trends. *Expert Syst. Appl.* 2014, 1118-1138.
- [19] Luo, W., Sun, J., Bu, C., Liang, H. Species – based Particle Swarm Optimizer enhanced by memory for dynamic optimization. *Appl. Soft Comput.* 2016, 47:130-140.
- [20] Mavrovouniotis, M., Li, C., Yang, S. A survey of swarm intelligence for dynamic optimization: algorithms and applications", *Swarm Evol. Comput.*, 2016.
- [21] Mavrovouniotis, M., Yang, S. A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. *Soft Comput.*, 2011.
- [22] Mavrovouniotis, M., Yang, S. Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors. *Appl. Soft Comput.* 2013, 13: 4023-4037.
- [23] Mavrovouniotis, M., Yang, S. Dynamic Vehicle Routing: A Memetic Ant Colony Optimization Approach. *Automated scheduling and planning* .2013, 283-301.
- [24] Mavrovouniotis, M., Yang, S. Ant algorithms with immigrant schemes for the dynamic vehicle routing problem. *Inform. Sciences.* 2015, 294:456-477.
- [25] Montoya-Torres, J., R., Lopez Franco, J., Nieto Isaza, S., Felizzola, H., J., Jimenez. A literature review on the vehicle routing problem with multiple depots. *Comput. Ind. Eng.* 2015, 79:115-129.
- [26] Nasiri, B., Meyabodi, M., R., Ebadzadeh, M., M. History-Driven Particle Swarm Optimization in dynamic and uncertain environments. *NeuroComputing.* 2016, 172: 356-370.
- [27] Nseef, S., K., Abdullah, S., Turkey, A., Kendall, G. An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems. *Knowl-based Syst.* 2016, 104: 14-23.
- [28] Pillac, V., Gendreau, M., Gueret, C., Medaglia, A., L.A review of dynamic vehicle routing problems. *Eur. J. Oper.*

- Res. 2013, 225:1-11.
- [29] Reed, M., Yiannakou, A., Evering, R. An ant colony algorithm for the multi-compartment vehicle routing problem. *Appl. Soft Comput.* 2014, 15: 169-176.
- [30] Schyns, M. An ant colony system for responsive dynamic vehicle routing. *Eur. J. Oper. Res.* 2015, 245: 704-718.
- [31] Yang, S. Genetic Algorithms with Memory- and Elitism-Based Immigrants in Dynamic Environments. *Evol. Comput.* 2008, 16: 385-416.
- [32] Yang, S., Jiang, Y., Thanh Nguyen, T. Metaheuristics for Dynamic Combinatorial Optimization Problems. *IMA J. Manage. Math.* 2011, 1 -29.
- [33] Yang, S., Tinos, R. A Hybrid Immigrants Scheme for Genetic Algorithms in Dynamic Environments. *Int. J. Autom. Comput.* 2007, 04: 243-254.
- [34] Neo Networking and Emerging Optimization Research Group. Capacitated VRP Instances.2013 [online]. Available at: <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/>
- [35] Talbi, E.,G., Metaheuristics from Design to Implementation, John Wiley & Sons, Inc., Hoboken, New Jersey. 2009.

Authors' Profiles



Dhanya K.M. received her B.Tech in Information Technology and M.Tech in Software Engineering from Cochin University of Science and Technology, Kerala. She worked as Lecturer in College of Engineering, Munnar, Kerala. She is Assistant Professor in Department of Information Technology, Government Engineering College, Palakkad, Kerala and currently doing research in Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry. She is Member of ISTE, India.



Dr. S. Kanmani received her B.E and M.E in Computer Science and Engineering from Bharathiyar University and Ph.D. from Anna University, Chennai. She had been the faculty of Department of Computer Science and Engineering, Pondicherry Engineering College from 1992 onwards. Presently she is Professor in Department of Information Technology, Pondicherry Engineering College, Puducherry. Her research interests are Software Engineering, Software Testing, Object-Oriented System, and Data Mining. She is Member of Computer Society of India, ISTE and Institute of Engineers, India. She has published about 50 papers in various International conferences and journals.

How to cite this paper: Dhanya K.M., S.Kanmani,"Dynamic Vehicle Routing Problem: Solution by Ant Colony Optimization with Hybrid Immigrant Schemes", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.9, No.7, pp.52-60, 2017. DOI: 10.5815/ijisa.2017.07.06