International Journal of Soft Computing 8 (5): 327-334, 2013
ISSN: 1816-9503

A New Hybrid Algorithm to Solve the Vehicle Routing Problem in the Dynamic Environment

[1]Messaoud Elhassania, [2]Boukachour Jaouad and [1]Elhilali Alaoui Ahmed
[1]Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, University Sidi Mohammed Ben Abdellah, Fez, Morocco
[2]Le Havre University Institute of Technology, Le Havre, France

Abstract: The Dynamic Vehicle Routing Problem (DVRP) is a natural extension of the classical Vehicle Routing Problem (VRP) which consists in designing routes for a fleet of capacitated vehicles that are to service a set of geographically dispersed points at the least cost. In DVRP, new customer demands are received along the day. Hence, they must be serviced at their locations by a set of vehicles in real time. In this research, to solve the DVRP, this later is decomposed into a series of static VRP and then a static solving algorithm which gives a quick and effectively good feasible solution in a given time is applied to each static VRP. This algorithm is a hybridization obtained by combining an Ant Colony Optimization (ACO) algorithm with a Large Neighborhood Search (LNS) algorithm. The computational experiments were applied to 22 benchmarks instances with up to 385 customers and the effectiveness of the proposed approach is validated by comparing the computational results with those earlier presented in the literature.

Key words: Vehicle routing problem, dynamic optimization, ant colony optimization, large neighborhood search, hybridization

INTRODUCTION

The Vehicle Routing Problem (VRP) is a well-known problem in operational research where customers of known demands are supplied by one or several depots. The objective is to find a set of delivery routes satisfying some requirements or constraints and giving minimal total cost. The VRP plays a vital role in distribution and logistics. Huge research efforts have been devoted to studying the VRP since 1959 where Dantzig and Ramser (1959) have described the problem as a generalized problem of Travelling Salesman Problem (TSP). Thousands of papers have been written on several VRP variants such as Capacitated Vehicle Routing Problem (CVRP) where each customer has a demand for a good and vehicles have finite capacity (Augerat *et al*., 1995; Baldacci *et al*., 2004; Chen *et al*., 2006; Kanthavel and Prasad, 2011). Vehicle Routing Problem with Time Windows (VRPTW) where each customer must be visited during a specific time frame (Thangiah *et al*., 1994; Cordeau *et al*., 2001; Braysy and Gendreau, 2005; Chen and Ting, 2005) and Vehicle Routing Problem with Pick-Up and Delivery (VRPPD) where goods have to be picked-up and delivered in specific amounts at the vertices (Min, 1989; Nagy and Salhi, 2005; Berbeglia *et al*., 2007; Gajpal and Abad, 2009), etc.

In the static version of the problem, it is assumed that all customers are known in advance to the planning process. However, it may be the case that customers, routing costs or service times become available in real-time once the service has begun. Due to the recent advances in communication technologies and positioning systems, it is now possible to address such dynamic problems.

The dynamic VRP with capacity and time duration constraints was introduced by Kilby *et al*. (1998) and further refined by Larsen (2000) and Montemanni *et al*. (2005). These researchers proposed some benchmark instances for the DVRP and presented a study on how the degree-of-dynamism affects the final travel costs. Montemanni *et al*. (2005) who extended Kilby *et al*. (1998) research, considered a DVRP as an extension to the standard VRP by decomposing a DVRP as a sequence of static VRPs and then solving them using an Ant Colony System (ACS) algorithm. Other optimization techniques have been applied. Branchini *et al*. (2009) presented an Adaptive granular local search heuristic. Creput *et al*. (2012) proposed the Self-Organizing Map (SOM) neural network into a population based evolutionary algorithm. Khouadjia *et al*. (2012) studied a Particle Swarm Optimization (PSO) and a Variable Neighborhood Search (VNS). As for static VRP, a lot of different versions of the

Corresponding Author: Messaoud Elhassania, Modeling and Scientific Computing Laboratory, Faculty of Science and Technology, University Sidi Mohammed Ben Abdellah, Fez, Morocco

DVRP have been studied. For example, the dynamic VRPTW is recognized as a standard problem well suited to allow comparative evaluations of heuristics and metaheuristics on a common set of benchmarks. To solve this problem, an evolutionary approach is proposed by Housroum *et al*. (2004) an Improved Large Neighborhood Search is introduced by Lianxi (2012) and an Ant Colony System is studied by Messaoud. Various other DVRP studies exist in the literature. Attanasio *et al*. (2004) give a Parallel Tabu Search heuristic for the dynamic multi-vehicle dial-a-ride problem. Finally, Gendreau *et al*. (2006) show proposes a neighborhood search heuristics to optimize the planned routes of vehicles in a context where new requests with a pick-up and a delivery location, occur in real-time, respectively.

The problem addressed in this study is the DVRP with capacity and time duration constraints only. In order to solve this problem, researchers divide the day in periods of equal duration. A request arriving during a time slice is not handled until the end of the time bucket thus during a time slice researchers only consider the requests known at its beginning. Hence, the optimization based on an Ant Colony Optimization (ACO) algorithm hybridized with a Large Neighborhood Search (LNS) algorithm is run statically during each time slice. The main advantage of this time partition is that similar computational effort is allowed for each time slice. This discretization is also possible by the nature of the requests which are never urgent and can be postponed.

THE DVRP

First, researchers show the event manager that get new orders, commits vehicles and constructs partial instances. Based on this manager, researchers propose a mathematic model for the DVRP.

Event manager: The event manager serves as an interface between the arrival of new orders and the optimization procedure. Based on the division of the working day into n_{ts} time slices (Fig. 1), each with an equal length of T/n_{ts} time where T is the length of the working day and researchers postpone the arrival of a new order to the end of that time slice so researchers can have an effective way to optimize new incoming orders over discrete time steps. This idea was first proposed by Kilby *et al.* (1998) and adopted by Montemanni *et al.* (2005). The goal is to bind the time given to each partial static problem hence providing an orderly way to service new requests. A different strategy is to include each new customer in the solution as soon as its request is received, it's may be necessary when urgent requests must be served as well as when customers have time windows constraints (Lianxi, 2012).

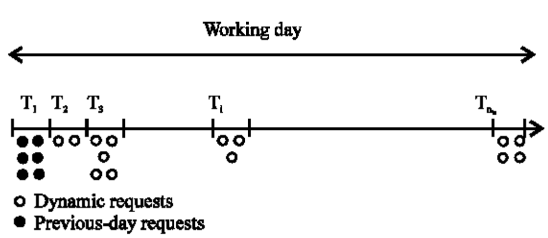

Fig. 1: The division of the working day

Each of time slices represents a partial static VRP where the vehicles must serve all known customers. The event manager runs in sequence the solving algorithm on these time slices. From the solutions provided by the algorithm, researchers decide about commitments within an advanced commitment time t_{ac} thus we allow a driver to react to new orders prior to the time of processing the order itself.

The first static problem created for the first time slice consists of all orders left over from the earlier working day. The time cut-off t_{co}, parameter controls the time in which new orders may arrive and thus may leave some customers unserved. These customers are carried over to the next working day. All the orders received after the t_{co} are interpreted as being customers that were not serviced the day before. This means that the optimization starts with customers who would have missed servicing yesterday because of the time cut-off.

The next static problem will consider all orders received during the earlier time slice as well as those which have not been committed to drivers yet. In the simulation, each vehicle m starts from the location of the last customer committed to it with a starting time corresponding to the end of the serving time for this customer and with a remaining capacity equal to the capacity left after serving all customers previously committed to vehicle m.

At the end of each time slice, the best solution is chosen and orders with a processing time (the processing time of an order starts when the vehicle assigned to it has to leave from its previous customer in order to travel to the next customer) starting within the next $T/n_{ts}+t_{ac}$ sec are committed to their respected vehicles. When any vehicle has used all its capacity, it is sent back to the depot.

As proposed by Montemanni *et al.* (2005), the pseudo-code of the event manager module is presented in Algorithm 1. The working day T is split into n_{ts} time slices, each one with T/n_{ts} duration. The set N_{Ti} where $l \in \{1, 2, ..., n_{ts}\}$ contains the orders known from the previous day if $l = 1$ and the orders received at the pervious time slice and the orders which have not been committed to drivers yet if $l \in \{2, ..., n_{ts}\}$. At the beginning of the working day the location of all the vehicles is set at the depot. A static problem is created in each loop step and solved with the procedure.

Algorithm 1 (Event Manger Procedure):
l: = 1;
T: = 0;
The starting time of each vehicle is set at the depot;
N_{T_l} := orders left the previous day;
While ($T<T_{co}$ or $N_{T_l} \neq \varnothing$)
StaticProblem:= problem with orders in N_{Tl}
Run the proposed approach (Algorithm 3) on StaticProblem;
CommOrd:= orders with processing time in $[T, T_{l+1}[$
Commit orders in CommOrd;
T: = T+T_l
l: = l+1;
N_{T_l} : = $N_{T_{l-1}}$ \CommOrd;
$N_{T_l} := N_{T_l} \bigcup \{\text{Orders appeared in} [T_{l-1}, T_l]\}$;
Update starting positions, capacities of vehicles and travel times;
EndWhile
Commit the depot to all the vehicles;
End

Mathematic model for DVRP: The dynamic vehicle routing problem is formulated in this study as a mixed integer linear programming problem. The objective of the formulation is to minimize the total travel time of the solution. The model seeks the optimization routes of every time slice which imply that the total distance is minimum. The DVRP can be described as follows.

Let G = (V', E) an undirected graph is given where V' = {0, 1, ..., n} is the set of n+1 vertices and E is the set of edges. Vertex 0 represents the depot and the vertex set V = V'\{0} corresponds to n customers. A non-negative cost d_{ij} and a travel time t_{ij} are associated with each edge $\{i, j\} \in E$; each vertex $v_i \in V$ has several non-negative weights associated with it, namely, a demand q_i and a service time s_i. A set M of identical vehicles of capacity Q at depot 0 must be used to visit the customers. A route is defined as a least cost simple cycle of graph G passing through depot 0 and such that the total demand of the vertices visited does not exceed the vehicle capacity and the maximum route duration is limited. Let $T = \{T_1, T_2, ..., T_{n_{ts}}\}$ is the division of the working day into n_{ts} time slices (Fig. 1). Where T_1 only considers the customers known from the earlier working day (i.e., the customers received after t_{co}). $T_i \backslash i \in \{2, ..., n_{ts}\}$ considers the customers received at the pervious time slice and those that have not been committed to drivers yet.

Let, L is a very large number, 0_{ml} is the last customer served by the vehicle m at the time slice T_{l-1}, Q^l_m is the total quantity ordered by customers already committed to vehicle m at the end of time slice T_{l-1} and D^l_m is the end of the serving time for the last customer committed to vehicle m at the time slice T_{l-1} (researchers assume $Q^l_m = D^l_m = 0$ for l = 1). The decision variables are defined as follows:

$$x^m_{ij} = \begin{cases} 1 & \text{if vehicle m drives from customer i to customer j} \\ 0 & \text{otherwise} \end{cases}$$

$$y^m_i = \begin{cases} 1 & \text{if vehicle m visit customer i} \\ 0 & \text{otherwise} \end{cases}$$

The DVRP can be formulated as the following integer program:

$$\text{Min} \sum_{m \in M} \sum_{i \in N_{T_l} \bigcup \{0,\, 0_{ml}\}} \sum_{j \in N_{T_l} \bigcup \{0,\, 0_{ml}\}} d_{ij} x^m_{ij} \quad \forall T_l \in T \tag{1}$$

$$\sum_{i \in N_{T_l}} x^m_{0_{ml} i} \le 1 \quad \forall \in M,\ \forall T_l \in T \tag{2}$$

$$\sum_{i \in N_{T_l}} x^m_{i0} \le 1 \quad \forall m \in M,\ \forall T_l \in T \tag{3}$$

$$\sum_{i \in N_{T_l}} y^m_i \le L. \sum_{j \in N_{T_l}} x^m_{0_{ml} j} \quad \forall m \in M,\ \forall T_l \in T \tag{4}$$

$$\sum_{i \in N_{T_l}} y^m_i \le L. \sum_{j \in N_{T_l}} x^m_{j0} \quad \forall m \in M,\ \forall T_l \in T \tag{5}$$

$$\sum_{m \in M} y^m_i = 1 \quad \forall i \in N_{T_l},\ \forall T_l \in T \tag{6}$$

$$\sum_{i \in N_{T_l}} x^m_{ij} = y^m_j \quad \forall m \in M,\ \forall j \in N_{T_l},\ \forall T_l \in T \tag{7}$$

$$\sum_{j \in N_{T_l}} x^m_{ij} = y^m_i \quad \forall m \in M,\ \forall i \in N_{T_l},\ \forall T_l \in T \tag{8}$$

$$Q^l_m + \sum_{i \in N_{T_l}} q_i y^m_i \le Q \quad \forall m \in M,\ \forall T_l \in T \tag{9}$$

$$D^l_m + \left(\sum_{i \in N_{T_l} \bigcup \{0,\, 0_{ml}\}} \sum_{j \in N_{T_l} \bigcup \{0,\, 0_{ml}\}} x^m_{ij} t_{ij} \right) + \sum_{i \in N_{T_l}} s_i y^m_i \le T \quad \forall m \in M,\ \forall T_l \in T \tag{10}$$

$$\sum_{i,j \in S} x^m_{ij} \le |S| - 1 \quad \forall m \in M,\ S \subset N_{T_l} \bigcup \{0,\, 0_{ml}\},\ 2 \le |S| \le |N_{T_l}| - 1,\ \forall T_l \in T \tag{11}$$

The objective function (1) is to minimize the total distance travelled during each time slice. Constraints of

Eq. 2-5 ensure that all vehicles start at the last customers have been committed to it and return to the depot for each time slice. Constraints of Eq. 6-8 ensure that each node except the depot is visited once and only once by one vehicle for each time slice. Constraint of Eq. 9 ensures that during each time slice, vehicle cannot exceed its capacity. The maximum route duration is limited by Eq. 10 for each time slice. The constraint condition of Eq. 11 removes the sub-loop for each time slice.

DVRP is obviously a NP hard problem (Larsen, 2000). In this study, researchers propose a new hybridization based on ACO algorithm and LNS algorithm to solve the DVRP up to 385 customers.

SOLVING DVRP

To solve the DVRP, a hybrid ACO algorithm with LNS algorithm is executed for each static VRP created at each time slice as described above. In this study, researchers adapt firstly the ACO algorithm for each static VRP and then researchers present the LNS algorithm to obtain the new hybridization for solving the DVRP.

The ACO algorithm: Marco Dorigo and colleagues introduced the first ACO algorithms in the early 1990's (Dorigo *et al*., 1991, 1996; Dorigo, 1992).

To solve the VRP for each time slice by the ACO algorithm, an individual ant constructs a solution by incrementally selecting customers until all customers have been visited. Initially, for the first time slice T_1, each ant k starts from the depot at the beginning of the day with a capacity Q. During the next time slices $T_l \backslash l \in \{2, 3, ..., n_{ts}\}$, each ant k starts initially at the location of $0_{m_1 T_l}$ (the last customer committed to the first vehicle $m_1 \in M$ used in the earlier time slice T_{l-1}). If the ant k reaches its capacity, it will move to the location of $0_{m_2 T_l}$ (the last customer served by the next vehicle $m_2 \in M$ used in the earlier time slice T_{l-1}), etc.

The starting time from the location $0_{mT_l} (m \in M)$ for each ant k, corresponds to the end of the serving time for the last customer committed to vehicle m while the capacity of the ant k will be equal to the residual capacity of the vehicle m, i.e., the capacity of the vehicle m minus the quantity ordered by customers already committed to vehicle m.

To select the next customer j for ant k currently at customer i, researchers use a greedy selection technique favoring cities with the best combination of short distance and large pheromone levels by Eq. 12:

$$j = \begin{cases} \arg\max_{u \in J_i^k} [\tau_{iu}^{\alpha} . \eta_{iu}^{\beta}] & \text{if} \quad q \le q_0 \\ J & \text{otherwise} \end{cases} \quad (12)$$

Where:

- q = A uniform random number in the interval [0, 1]
- q_0 = The parameter which determines the relative importance of exploitation versus exploration
- J_i^k = The set of cities that have not yet been visited by ant k
- α = A parameter which determine the relative influence of heuristic information
- β = The importance of distance in comparison to pheromone quantity
- $\eta_{ju} = 1/d_{ju}$ = The attractiveness of the move as computed by some heuristic information, indicating a priori desirability of that move
- τ_{ij} = The pheromone trail level of the move, indicating how profitable it has been in the past to make that particular move
- $J \in J_i^k$ = Choose according to the probability by Eq. 13:

$$P_{iJ}^k = \frac{\tau_{iJ}^{\alpha} . \eta_{iJ}^{\beta}}{\sum_{h \in J_i^k} \tau_{ih} . \eta_{ih}^{\beta}} \quad (13)$$

The pheromone level on the selected edge is updated according to the local updating rule in Eq. 14.

$$\tau_{ij}(t+1) = (1-\rho).\ \tau_{ij}(t) + \rho.\tau_0 \quad (14)$$

Where:

- ρ = A parameter that controls the evaporation of the pheromone trail ($0<\rho<1$)
- τ_0 = The initial amount of pheromone deposited on each of the edges:

$$\tau_0 = \frac{1}{\text{Distance (bestknow)}}$$

where bestknow is the best solution found in the static version of the problem

Upon conclusion of an iteration (i.e., once all ants have constructed a tour), global updating of the pheromone takes place. Edges that compose the best solution are rewarded with an increase in their pheromone level which is expressed in Eq. 15:

$$\tau_{ij}(t+1) = (1-\rho).\ \tau_{ij}(t) + \rho.\Delta\tau_{ij}(t)$$
$$\Delta\tau_{ij} = \frac{1}{L^+} \quad (15)$$

where, L^+ is the length of the best (shortest) tour found so far.

The LNS algorithm: To improve the quality of solution found for each time slice, the ACO is mixed with the LNS proposed by Shaw (1998). Researchers explore a large neighborhood of current solution by selecting a number of nodes to remove from the routing plan and re-inserting them to the solution. This metaheuristic starts with an initial solution and iteratively a set of requests are removed from their tours in the current solution. Then, a new solution is generated by including the removed requests and the new solution is accepted as the next current solution if the total distance of the new solution decreases.

Removal algorithm: Random removal selects a fixed number of requests randomly and removes them from the solution. This obviously has the effect of diversifying the search. Related removal removes a set of requests that in some sense are related and hence easy to interchange. For the VRP, researchers define the relatedness R_{ij} of two orders i and j solely by the distance between the requests d_{ij} and v_{ij} defined as follows:

$$\upsilon_{ij} = \begin{cases} 1 & \text{if i and j are assigned in the same vehicle} \\ 0 & \text{otherwise} \end{cases}$$

The relatedness measure used to remove customers is:

$$R_{ij} = \frac{1}{\upsilon_{ij} + d_{ij}}$$

This algorithm, initially selects randomly a request i. Then, it repeatedly chooses an already selected request j and selects a new request which is most related to j. The algorithm stops when a fixed number of requests have been chosen.

Re-inserting algorithm: The customers are inserted into the position that minimizes the insertion cost over all the routes.

Algorithm 2 (The large neighborhood search approach):

```
The LNS Procedure (Solution S)
t: = 0;
While (t < T/(15×n_ts))
Select R/2 customers by random removal
Select R/2 customers by related removal
Remove the selected customers from S to obtain a partial solution S1
Apply the re-inserting algorithm on S1 to obtain the new solution S'

If (distance (S')<distance (S))

s: = s'
EndIf
Update the time t
EndWhile
End
```

Researchers show the pseudo-code of the hybrid algorithm named ACOLNS:

Algorithm 2 (The proposed approach ACOLNS for the DVRP):

```
The ACOLNS procedure (StaticProblem P)
For each arc (i, j)
τij = τ0
EndFor
While (time < T/n_ts)
  For k = 1 to ant_number
  Construct a current_solution for P according to
section of the ACO algorithm
   If (k = 1)
     Else
     If (distance (Saide)>distane(current_solution))
      EndIf
    EndIf
  EndFor
   Saide: = LNS (Saide);
   If (distance(Saide)<distance(Sglobal))
   Sglobal: = Saide
  EndIf
  For each move (i, j) in solution Sglobal
  Update the trace level τij according to (15)
  EndFor
  Update time
EndWhile
```

EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed approach, this later is tested on the benchmark data set proposed by Kilby *et al.* (1998) and extended by Montemanni *et al.* (2005). There are 22 benchmarks instances derived from the conventional available VRP benchmark data, namely Taillard (1994) (13 benchmarks instances), Christofides and Beasley (1984) (7 benchmarks instances) and Fisher and Jaikumar (1981) (2 benchmarks instances). The number of customers ranges in (50, 385) and the service area may consist of uniformly distributed customers, clustered customers or a combination of both.

The cut-off time t_{co} and the advanced commitment time t_{ac} are set to T/2 and 0, respectively. The total length of the working day T is 1500 sec and according to Montemanni *et al.* (2005) the best number of time slices n_{ts} is 25. The algorithm is written in C++ and was run on a MacBook Pro-Core i5/2.4 GHz-MacOS X 10.7 Lion. The setting parameters shown below give a good compromise between the computation time and the solution quality (Table 1).

Table 1: ACOLNS parameters

Parameters	Values
ρ	0.1
q_0	0.9
ant_number	3.0
α	1.0
β	2.0
R	15.0

According to the objective function 1, Table 2 shows the best and the average total travel distance. The results are obtained over 5 runs using ACO algorithm alone and then using ACO algorithm with LNS algorithm. Researchers can see from Table 2 and Fig. 2 that ACOLNS algorithm outperforms ACO algorithm. In fact, the ants guide the search process into promising regions of the solution space where the LNS algorithm can find good solutions.

Table 2: Results obtained by ACO and ACOLNS

	ACO		ACOLNS	
Results	Best	Average	Best	Average
c50	637.94	659.60	601.78	623.09
c75	1078.35	1101.03	1003.20	1013.47
c100	1065.38	1120.31	987.65	1012.30
c100b	916.25	972.30	932.35	943.05
c120	1410.98	1481.27	1272.65	1451.60
c150	1530.06	1559.00	1370.33	1394.77
c199	1832.87	1892.48	1717.31	1757.02
tai75a	2197.79	2215.75	1832.84	1880.87
tai75b	1722.22	1821.00	1456.97	1477.15
tai75c	2029.29	2076.19	1612.10	1692.00
tai75d	1600.57	1748.09	1470.52	1491.84
tai100a	2842.54	2863.71	2257.05	2331.28
tai100b	2666.74	2750.01	2203.63	2317.30
tai100c	1930.64	2051.87	1660.48	1717.61
tai100d	2352.84	2410.37	1952.15	2087.96
tai150a	3848.24	4191.37	3436.40	3595.40
tai150b	3588.04	3775.38	3060.02	3095.61
tai150c	3577.68	3733.77	2735.39	2840.69
tai150d	3667.74	3745.31	3138.70	3233.39
tai385	33645.85	34885.81	33062.06	35188.99
f71	340.30	358.45	311.33	320.00
f134	17473.07	17983.84	15557.82	16030.53

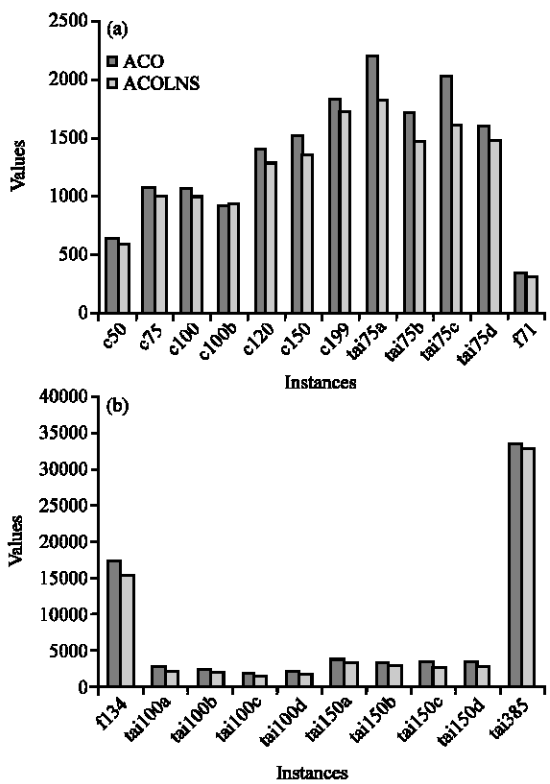

Fig. 2a, b: ACO compared with ACOLNS

The following study aims to compare the results with those obtained by GRASP (Greedy Randomized Adaptive Search Procedure) where the objective is to repeatedly sample stochastically greedy solutions and then use a local search procedure to refine them to a local optima. The strategy of the procedure is centered on the stochastic and greedy step-wise construction mechanism that constrains the selection and order-of-inclusion of the components of a solution based on the value they are expected to provide.

ACS (Ant Colony System) algorithm which is inspired by the foraging behavior of ants, specifically the pheromone communication between ants regarding a good path between the colony and a food source in an environment.

DAPSO (Dynamic Adapted Particle Swarm Optimization) algorithm which follows a collaborative population based search which models over the social behavior of bird flocking and fish schooling.

VNS (Variable Neighborhood Search) which is characterized by the usage of one single solution at each time, this metaheuristic is based on the principle of systematically changing neighborhoods in order to escape from local optima.

The '-' means that the corresponding problem is not tested or a feasible solution cannot be obtained in the literature. A comparison of the solution quality in terms of minimizing travel distances is done between the approach and other metaheuristics proposed earlier in literature. These metaheuristics are GRASP and ACS proposed by Montemanni *et al.* (2005) and DAPSO and VNS proposed by Khouadjia *et al.* (2012).

As researchers can see the approach is very competitive. It outperforms Montemanni's Grasp and Montemanni's ACS on 18 and 15 benchmarks instances, respectively. Furthermore, it outperforms Khouadija's DAPSO and Khouadija's VNS on 7 and 8 benchmarks instances, respectively. The proposed approach gives new best-so-far-solutions on benchmarks instances c120 and tai100d and it finds a feasible solution unlike the other approaches of literature which are not tested or feasible solutions cannot be obtained for the tai385 benchmark instance. The approach provides also the shortest average for the travelled distance on 15 benchmarks instances. All these results allow us to say that the hybridization is effective and shows the viability to generate very high quality solutions for the DVRP (Table 3).

Table 3: Numerical results for ACOLNS compared to other metaheuristics

	GRASP		ACS		DAPSO		VNS		ACOLNS	
Results	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average
c50	696.92	719.56	631.30	681.86	575.89	632.38	599.53	653.84	601.78	623.09
c75	1066.59	1079.16	1009.38	1042.39	970.45	1031.76	981.64	1040.00	1003.20	1013.47
c100	1080.33	1119.06	973.26	1066.16	988.27	1051.50	1022.92	1087.18	987.65	1012.30
c100b	978.39	1022.12	944.23	1023.60	924.32	964.47	866.71	942.81	932.35	943.05
c120	1546.50	1643.15	1416.45	1525.15	1276.88	1457.22	1285.21	1469.24	1272.65	1451.60
c150	1468.36	1501.35	1345.73	1455.50	1371.08	1470.95	1334.73	1441.37	1370.33	1394.77
c199	1774.33	1898.20	1771.04	1844.82	1640.40	1818.55	1679.65	1769.95	1717.31	1757.02
tai75a	1911.48	2005.44	1843.08	1945.20	1816.07	1935.28	1806.81	1954.25	1832.84	1880.87
tai75b	1582.24	1758.88	1535.43	1704.06	1447.39	1484.73	1480.70	1560.71	1456.97	1477.15
tai75c	1596.17	1674.37	1574.98	1653.58	1481.35	1664.40	1621.03	1746.07	1612.10	1692.00
tai75d	1545.21	1588.73	1472.35	1529.00	1414.28	1493.47	1446.50	1541.98	1470.52	1491.84
tai100a	2427.07	2510.29	2375.92	2428.38	2249.84	2370.58	2250.50	2462.50	2257.05	2331.28
tai100b	2302.95	2512.27	2283.97	2347.90	2238.42	2385.54	2169.10	2319.72	2203.63	2317.30
tai100c	1599.19	1704.40	1562.30	1655.91	1532.56	1627.32	1490.58	1557.81	1660.48	1717.61
tai100d	1973.03	2087.55	2008.13	2060.72	1955.06	2123.90	1969.94	2100.38	1952.15	2087.96
tai150a	3787.53	3899.16	3644.78	3840.18	3400.33	3612.79	3479.44	3680.35	3436.40	3595.40
tai150b	3313.03	3485.79	3166.88	3327.47	3013.99	3232.11	2934.86	3089.57	3060.02	3095.61
tai150c	3110.10	3219.27	2811.48	3016.14	2714.34	2875.93	2674.29	2928.77	2735.39	2840.69
tai150d	3159.21	3298.76	3058.87	3203.75	3025.43	3347.60	2954.64	3147.38	3138.70	3233.39
tai385	-	-	-	-	-	-	-	-	33062.06	35188.99
f71	359.16	376.66	311.18	348.69	279.52	312.35	304.32	325.18	311.33	320.00
f134	15433.84	16458.47	15135.51	16083.56	15875.00	16645.89	15680.05	16522.18	15557.82	16030.53

CONCLUSION

This research studies the dynamic vehicle routing problem with capacity and time duration constraints which is decomposed into a serial static VRP. A model mathematic and a new hybridization based on ACO algorithm and LNS algorithm is proposed to solve the problem. The computational results show that the proposed approach is able to generate satisfactory solution in simulated dynamic environment. As for future research, it may be interesting to test this approach with other variants of DVRP.

REFERENCES

Attanasio, A., J.F. Cordeau, G. Ghiani and G. Laporte, 2004. Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Comput., 30: 377-387.

Augerat, P., J.M. Belenguer, E. Benavent, A. Corberan, D. Naddef and G. Rinaldi, 1995. Computational results with a branch and cut code for the capacitated vehicle routing problem. Research Report 949-M, Universite Joseph Fourier, Grenoble, France.

Baldacci, R., E. Hadjiconstantinou and A. Mingozzi, 2004. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. Oper. Res., 52: 723-738.

Berbeglia, G., J.F. Cordeau, I. Gribkovskaia and G. Laporte, 2007. Static pickup and delivery problems: A classification scheme and survey. TOP, 15: 1-31.

Branchini, R.M., V.A. Armentano and A. Lokketangen, 2009. Adaptive granular local search heuristic for a dynamic vehicle routing problem. Comput. Oper. Res., 36: 2955-2968.

Braysy, O. and M. Gendreau, 2005. Vehicle routing problem with time windows, Part II: Metaheuristics. Transport. Sci., 39: 119-139.

Chen, A.L., G.K. yang, and Z.M. Wu, 2006. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. J. Zhejiang Uni. Sci., 7: 607-614.

Chen, C.H. and C.J. Ting, 2005. A hybrid ant colony system for vehicle routing problem with time windows. J. East. Asia Soc. Transp. Stud., 6: 2822-2836.

Christofides, N. and J.E. Beasley, 1984. The period routing problem. Networks, 14: 237-256.

Cordeau, J.F., G. Laporte and A. Mercier, 2001. A unified tabu search heuristic for vehicle routing problem with time windows. J. Operat. Res. Soc., 52: 928-936.

Creput, J.C., A. Hajjam, A. Koukam and O. Kuhn, 2012. Self-organizing maps in population based metaheuristic to the dynamic vehicle routing problem. J. Combinatorial Optim., 24: 437-458.

Dantzig, G.B. and J.H. Ramser, 1959. The truck dispatching problem. Manage. Sci., 6: 80-91.

Dorigo, M., 1992. Optimization, learning and natural algorithms. Ph.D. Thesis, Department of Electronics, Politecnico di Milano, Italy.

Dorigo, M., V. Maniezzo and A. Colorni, 1991. Positive feedback as a search strategy. Technical Report No. 91-016, Politecnico di Milano, Italy.

Dorigo, M., V. Maniezzo and A. Colorni, 1996. Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B: Cybern., 26: 29-41.

Fisher, M.L. and R. Jaikumar, 1981. A generalized assignment heuristic for vehicle routing. Networks, 11: 109-124.

Gajpal, Y. and P. Abad, 2009. An ant colony system for the vehicle routing problem with simultaneous delivery and pickup. Comput. Operat. Res., 36: 3215-3223.

Gendreau, M., F. Guertin, J.V. Potvin and R. Seguin, 2006. Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. Transport. Res. Part C, 14: 157-174.

Housroum, H., T. Hsu, R. Dupas and G. Goncalves, 2004. An hybrid ga approach for solving the dynamic vehicle routing problem with time windows. Proceeding of the ICTTA '06. 2nd, Information and Communication Technologies, 2006, April, 24-28, 2006, Damascus, pp: 787-792.

Kanthavel, K. and P. Prasad, 2011. Optimization of capacitated vehicle routing problem by nested particle swarm optimization. Am. J. Appl. Sci., 8: 107-112.

Khouadjia, R.M., B. Sarasola, E. Alba, L. Jourdan and E.G. Talbi, 2012. A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests. Appl. Soft Comput., 12: 1426-1439.

Kilby, P., P. Prosser and P. Shaw, 1998. Dynamic VRPs: A study of scenarios. Technical Report APES-06-1998, University of Strathclyde, UK.

Larsen, A., 2000. The dynamic vehicle routing problem. LYNGBY, IMM-PHD-73. Technical University of Denmark (DTU).

Lianxi, H., 2012. An improved LNS algorithm for real-time vehicle routing problem with time windows. Comput. Opreat. Res., 39: 151-163.

Min, H., 1989. The multiple vehicle routing problem with simultaneous delivery and pick up points. Transport. Res. part A-Policy pract., 23: 377-386.

Montemanni, R., L.M. Gambardella, A.E. Rizzoli and A.V. Donati, 2005. Ant colony system for a dynamic vehicle routing problem. J. Comb. Optim., 10: 327-343.

Nagy, G. and S. Salhi, 2005. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. Euro. J. Operat. Res., 162: 126-141.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems. Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming, October 26-30, 1998, Pisa, Italy, pp: 417-431.

Taillard, E., 1994. Parallel iterative search methods for vehicle routing problems. Networks, 23: 661-673.

Thangiah, S., I.H. Osman and T. Sun, 1994. Hybrid Genetic Algorithm Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows. Working Paper UKC/IMS/OR94/4 Institute of Mathematics and Statistics. University of Kent, Canterbury.