Ant Colony System for a Problem in Reverse Logistic

Franklin JOHNSON[1]**, Jorge VEGA**[2]**, Guillermo CABRERA**[3,*]**, Enrique CABRERA**[4]

[1] Universidad de Playa Ancha, CHILE,
franklin.johnson@upla.cl

[2] Universidad de Antofagasta, Department of Electrical Engineering, CHILE,
jorge.vega@uantof.cl

[3] Pontificia Universidad Católica de Valparaíso, CHILE,
guillermo.cabrera@ucv.cl

[4] Universidad de Valparaíso, CIMFAV, CHILE,
enrique.cabrera@uv.cl

* *coresponding author*

Abstract: Distribution, redistribution, recycling and repacking have become an important issue in logistic planning during the last decades. While keeping operational cost as low as possible still the main goal for logistic planners, other aspects such as recycling are getting more attention from industry. In this article the well known Ant Colony System (ACS), a bio-inspired algorithm, is implemented to solve a problem arising in Reverse Logistic namely Vehicle Routing Problem with Simultaneous Delivery and Pickup (VRPSDP). To solve this problem we need to find the optimal set of paths that meet, at the same time, customer delivery and pickup demands. In order to solve this problem, our ACS implementation makes use of a strategy that mimics the effect of the pheromone in the natural Ants behaviour. To do that, each vehicle is viewed as an individual agent (ant) and consequently its behaviour is driven by pheromone strategy, i.e. it tends to choose the route for which the pheromone level is higher. Results show that our ACS implementation provides good quality solutions within an acceptable time. Furthermore, obtained solutions are quite competitive when compared to other stochastic techniques previously studied in literature.

Keywords: vehicle routing problem with simultaneous delivery and pickup, ant colony system, reverse logistic.

1. Introduction

During the last few decades an increasing number of environmental issues have led to changes in the normative that regulates the logistic industry. In particular, new requirements from customers such as recycling and proper items disposal cause that logistic industry should not only look at the delivery process but also to the pick-up one. Because of that, well-known Vehicle Routing Problem (VRP) is not an adequate model to the integrated problem mentioned before. Therefore, several new models have been introduced in order to address this new scenario. In this article we consider one of those models namely VRP with simultaneous delivery and pick-up (VRPSDP). VRPSDP problem has been firstly introduced and modelled in [1]. In [1] authors studied the VRPSDP based on a real case from book distribution industry. They tackled the book logistic activity from one central library to a set of local libraries and vice-versa. They considered a fixed number of vehicles and a limited vehicle capacity. Different mathematical models for the VRPSDP have been proposed in [2], [3] and [4]. Particularly, authors in [2] modelled the VRPSDP as part of the reverse logistics process. In this paper we attempt to find high quality solutions to this optimisation problem using the well-known Ant Colony Systems (ACS) heuristic.

Ant Colony Systems has demonstrated to be very effective for routing problems [9]. In general it is able to find near-optimal routes for many problems such as VRP [14, 15, 20] and travelling salesman problem [10, 12, 16], among others. One advantage of ACS over other heuristics is its rapid convergence, which means high quality solutions within an acceptable time. To the best of our knowledge, ACS algorithm has only been used to solve the VRPSDP problem in [15, 21]. Although similar strategies are considered, our algorithm implements quite different steps and rules which make our approach substantially different from those previously presented approaches. This article is organised as follows: Section 2 presents a literature review and introduces the mathematical model for the VRPSDP that is considered in this study. Section 3 reviews ACS algorithms and describes in detail the proposed ACS algorithm for the VRPSPD. Section 4 presents the benchmark used in this study. Obtained results are analysed at the end of this section. Finally, some conclusions are outlined in Section 5.

2. Literature Review

As mentioned before, the VRPSDP was firstly introduced in [1]. After that, several authors have tackled this problem making use of different techniques. The VRPSDP problem is solved by means of an insertion-based heuristic in [2]. Authors in [3] present the VRP with pickup and delivery (VRPPD), the non-simultaneous version of the VRPSDP. They avoid traditional insertion strategies and, instead, propose a strategy that considers both items delivery and pickups simultaneously. Their strategy generates, as a first step, a solution to the simplified VRP. Then needed modifications on the obtained solutions are applied in order to make these solutions feasible for the main problem. In [5] authors consider the well-known bio-inspired algorithm particle swarm optimization (PSO). Authors highlight rapid convergence of PSO over other heuristic methods. Moreover, in [6] authors propose a local search heuristic to solve the VRPSPD problem that makes some improvements on the tour partitioning heuristic commonly applied to the well-known travelling salesman problem. In [7] a hybrid algorithm that combines two heuristics approaches that have shown to be very efficient on solving routing problem: guided local search and Tabu Search. The main goal of this hybrid strategy is to keep a good balance between exploration and exploitation of promising regions. Tabu Search algorithm is also considered in [4] to solve a variant of the VRPSPD problem which includes a constraint which bounds the maximum travel distance. More recently, authors in [18] implement a Genetic Algorithm to solve the multi-objective version of the VRPSDP problem where trade-off between transportation costs and vehicles needed to solve the problem is considered. In [19] authors propose a novel improved differential evolution algorithm. They claim that their differential evolution approach performs better than other well-known Genetic Algorithms proposed in the literature.

Exact techniques have also been used. In [17] the authors propose a cutting-plane-based technique which allows them to find optimal solutions for medium size instances. This kind of techniques becomes very expensive in terms of computational time as the problem size increases though.

Several mathematical models have been proposed in the literature to the VRPSDP. Particularly, in [7] three different models involving the items distribution from a central depot to a set of customers as well as from customers back to a central depot have been studied. These models are: VRP with backhauls (VRPB), VRP with mixed pick-ups and deliveries (VRPMPD) and, finally, the VRPSDP that is considered in this paper. We briefly describe these three models with the focus on the VRPSDP model. Interested readers are referred to [7] for further details on these three problems.

VRPB is also known as delivery first pick-up second VRP. Here customers are firstly grouped to a set of line-hauls customers, i.e. those customers that demand delivery services, and a set of backhauls customers, i.e. those that demand pick-up services. Line- and back-hauls customer demands need to be satisfied by mean of a set of vehicles with a limited capacity which operates between a central depot and customers. Vehicles have to, firstly, satisfy the line-hauls customers and, after that, visit the back-hauls customers to get the items back to the central depot. This model considers this precedence restriction because of limitations related to re-arrangement of items during delivery and pick-up activity. Thus, the vehicles under this model have to, firstly, deliver all their carried items and then pick-up items from back-hauls customers. Under this model study of the capacity constraint becomes easier, because of the fact that vehicles would always meet the maximum allowed load assuming that both total delivery and pick-up items are below vehicles capacity.

Similar to VRPB problem, VRP with mixed pick-ups and deliveries (VRPMPD) problem allows customers to either receive or take items back to a central depot. Opposite to the problem described above (VRPB), precedence restriction is not required, i.e. order in which deliveries and pick-ups occur is not important under this model. This is possible because under this model we can take advantage of rear- and side-loading doors of vehicles which allows us to include new items on the vehicle load without rearranging already existing items on the vehicle. The fact that precedence constraint is no longer required allows us to obtain solutions that are more useful for real cases arising in industry. Unfortunately, maximum load restriction makes the problem harder as deliveries and pick-ups are scheduled at any order and consequently the load of the vehicles changes after each customer is served.

Thus, in order to define whether a particular route satisfies the vehicle capacity constraint, we need to evaluate the vehicle load at each arc that is part of the route, and check if it meets the vehicle capacity limit.

Finally, the third model we present in this section is the one we are focused on this study, the VRP with simultaneous deliveries and pick-ups. This problem can be seen as a generalization of the problem mentioned before (VRPMPD) or, alternatively, the VRPMPD can be considered as a VRPSDP case where either delivered or picked-up items is equal to zero. In VRPSDP case, clients could ask for simultaneous pick-up and delivery services. This has some implications on the behaviour of the vehicle load. While for the VRPB case number of items carried by a vehicle decreases to zero and then increases to some level below the vehicle capacity, in the VRPMPD the vehicle load increases and decreases in turns, i.e. the vehicle delivers items to customer *i* and then picks items up from customer *i*+1. A different situation occurs when VRPSPD is considered. In this case items are delivered and picked up at the same time. We do not know a priori whether the difference between delivered and picked items would be positive or negative. Thus, we have one situation where a) the vehicle load only increases after certain number of customers, b) vehicle loads only decreases after certain number of customers or, a mix of situations a and b above. Clearly, this situation makes the problem much harder to solve than previously presented models. Below, we present the mathematical formulation for VRPSDP obtained from [2].

Let J be the set of all customer and $J_0 = J \cup \{0\}$ be the set of all customer and a central depot. Total number of nodes (number of customers + 1) is denoted by $n = |J_0|$. Let D_j be the number of items to be delivered to customer $j \in J$ and P_j be the number of items to be picked up from customer $j \in J$. Let V be the set of all the considered vehicles. Let I^{cap} be the vehicle capacity. We assume all vehicles have same capacity. Distance between nodes *i* and *j* is denoted by c_{ij}, where $i \in J_0$, $j \in J_0$ and $i \neq j$. If $i = j$ then $c_{ii} = M$, where M is a large number such as:

$$M = max\left\{\sum_{j \in J} (D_j + P_j), \sum_{i \in J} \sum_{j \in J_0,\, i \neq j} C_{ij}\right\}.$$

Distance $c_{00} = 0$.

The model for the VRPSDP is as follows:

$$minimise\, z = \sum_{i \in J_0} \sum_{j \in J_0} \sum_{v \in V} c_{ij} \times x_{ijv} \tag{1}$$

Subject to

$$\sum_{i \in J_0} \sum_{v \in V} x_{ijv} = 1 \qquad \forall j \in J \tag{2}$$

$$\sum_{i \in J_0} x_{isv} = \sum_{j \in J_0} x_{sjv} \qquad \forall s \in J, \forall v \in V \tag{3}$$

$$l_v^0 = \sum_{i \in J_0} \sum_{j \in J} D_j \times x_{sjv} \quad \forall v \in V \tag{4}$$

$$l_v^1 \geq l_v^0 - D_j + P_j - M\left(1 - x_{0jv}\right) \quad \forall i, j \in J \tag{5}$$

$$l_v^j \geq l_v^i - D_j + P_j - M\left(1 - \sum_{v \in V} x_{ijv}\right) \quad \forall i, j \in J, i \neq j \tag{6}$$

$$l_v^k \leq I^{cap} \qquad \forall k = \{0, \ldots, n\}, \forall v \in V \tag{7}$$

$$\pi_j \geq \pi_i + 1 - n\left(1 - \sum_{v \in V} x_{ijv}\right) \quad \forall i, j \in J, i \neq j \tag{8}$$

$$\pi_j \geq 0 \forall j \in J \tag{9}$$

$$x_{ijv} \in \{0,1\} \forall i, j \in, \forall v \in V \tag{10}$$

Objective function (1) is the total distance travelled by all the considered vehicles. Constraint (2) makes sure that all customers are served exactly once. Constraint (3) ensures that each customer is always served by the same vehicle (forward and reverse logistic). Constraint (4) sets the starting vehicle loads and inequalities (5-6) limit the vehicle load after visiting customer one and vehicle loads for all other arcs considered within the route, respectively. We can also limit the vehicle loads as in (7). Equation (8) makes sure that no sub-tours are allowed. Finally, Equations (9-10) are integrity constraints.

3. Proposed ACS approach.

In this section we begin with a general explanation on the Ant System (AS) algorithm which is the framework of the Ant Colony System implemented in this study.

AS metaheuristic is based on real ants behaviour. One distinctive feature in AS is the indirect communication between so called artificial ants by mean of artificial pheromone trails. This pheromone trails in AS corresponds to distributed, numerical data that is used by the artificial ants to decide among alternative paths to generate a solution for the problem that is being solved [8].

Basically, AS extracts three main ideas from natural ant behaviour [10]: (i) An individual ant prefers those paths which accumulate higher pheromone level, (ii) The shorter the path the faster pheromone level increases; and (iii) Ants communication is done by means of pheromone trails.

We can describe AS strategy as follows: A colony of independent artificial ants that moves through states of the problem in a concurrent and asynchronous way generating partial solutions of the optimisation problem. This artificial ants move from one state to another by means of a probabilistic local decision rule called *state transition rule* that is based on two parameters: trails and attractiveness. This policy assumes that artificial ants prefer to move through paths that concentrate higher pheromone level. After each iteration, each ant builds up a solution. After the last iteration, each ant evaluates the solution and modifies the pheromone level on the paths that are part of its solution. Changing pheromone levels is important as future ants will make their decision among different paths based on this information.

Moreover, *trail evaporation* and, optionally, *daemon actions* are two additional features of AS. The first one decreases all pheromone levels at each iteration to limit pheromone levels on an arc. The second mechanism that could be used when implementing an AS algorithm is the so called daemon actions. This mechanism might be considered to implement centralized actions that are not possible to be implemented by a single ant. These centralized actions includes, among others, a call to a local optimization strategy and global information (such as pheromone levels) updating process, to decide whether to bias the search process from a non-local view.

Another mechanism that is implemented in AS is the *move probability distribution* which defines probability $p_k(r,s)$ as the probability of an ant *k* in *r* chooses s as its next node. This probability would be equal to 0 if leg (*r*,*s*) is not feasible for ant *k*. A move might be not feasible if, for instance, source and destination nodes are both in the tabu list of ant *k*. Tabu list corresponds to a list that includes all those infeasible moves for ants k starting from state *r*. If a move is feasible, its probability is computed as in Equation (11).

$$p_k(r,s)=\begin{cases}\dfrac{[\tau(r,s)][\eta(r,s)]^{\beta}}{\sum\limits_{u\in tabu_k}[\tau(r,u)][\eta(r,u)]^{\beta}} & if\, s\notin tabu_k\\ 0 & otherwise\end{cases} \tag{11}$$

In expression (11), *tabu*$_k$ is the tabu list of ant k explained before. Parameter β corresponds to a number that is a coefficient between pheromone levels and distance. At the end of each iteration *t*, i.e. after every ant has generated its solution, pheromone levels are modified using the following rule:

$$\tau(r,s)=(1-\alpha)\,\tau(r,s)+\sum_{k=1}^{m}\Delta\,\tau_k(r,s) \tag{12}$$

where

$$\Delta\,\tau_k(r,s)=\begin{cases}\dfrac{1}{L_k} & if\,(r,s)\in tour\, done\, by\, ant\, k\\ 0 & otherwise\end{cases}$$

corresponds to the cumulative contributions of ants that pass through arc (r,s) to generate its solutions. *Evaporation coefficient*, denoted by α with 0≤ α ≤1, is a parameter defined by the user. Length of the tour performed by ant *k* is denoted by L_k. Finally, *m* corresponds to the total number of ants.

In order to make those paths that are better in terms of the objective function value more attractive for future ants, contribution of each ant is proportional to the quality of its solutions. In this way, arcs that generate better solutions will have higher pheromone levels and, consequently, will be more attractive to future ants. This is also important for the algorithm convergence speed.

AS algorithms including all mechanism described before have demonstrated to be very effective in solving complex routing optimisation problems such as the well-known travelling salesman problem and VRP (see [8] and [9] respectively for further details).

Ant Colony System (ACS) has three variants w.r.t. the AS algorithm [12] explained above. Firstly, ACS incorporates a state transition rule that allows us to keep a balance between exploration of new edges and exploitation of a priori and accumulated knowledge about the problem. Moreover, we apply the global updating rule only to the best tour in ACS. This means that pheromone levels in those arcs that are not part of the best tour can only decrease due to the trail evaporation mechanism. Finally, ACS implements a local updating rule that is applied at each step of the solution generation. Equations (13), (14) and (15) show these variants.

$$s=\begin{cases} \underset{u\notin tabu_k}{argmax}\,[\tau(r,u)][\eta(r,u)]^{\beta} & if\ q\leqslant q_0 \\ S & otherwise \end{cases} \qquad (13)$$

Where q is a pseudo-random number uniformly distributed [0, 1] and q_0 is a user defined parameter, with $0\leqslant q_0\leqslant 1$. On the one hand parameter q_0 close to 0 means that more exploration over the search space would be performed. On the other hand, parameter q_0 near to 1 means more exploitation over a specific neighbourhood. S is a random value.

$$\tau(r,s)=(1-\alpha)\tau(r,s)+\alpha\Delta\tau_k(r,s) \qquad (14)$$

Where

$$\Delta\tau(r,s)=\begin{cases} \frac{1}{L_{gb}} & if\ (r,s)\in global\ best\ tour \\ 0 & otherwise \end{cases}$$

L_k is the global best tour length. A local updating rule is also implemented in ACS algorithm. This local rule is applied at each iteration and is as follows,

$$\tau(r,s)=(1-\rho)\,\tau(r,s)+\rho\Delta\,\tau(r,s) \qquad (15)$$

where 0 ≤ ρ ≤1. Finally, Algorithm 1 shows the ACS structure.

Algorithm 1: Ant_Colony_System Algorithm Framework

```
Begin{
 Do{
  For each ant: initialize its solution
   Do{
    For each ant:
     Apply rule (η(r,s),τ(r,s))
     Local Update
    End for
   }Until all ants have completed their solutions
   Global_update $(\tau(r,s))$
  }Until stop criterion is reached
}End
```

4. Computational Results

In this paper an ACS heuristic has been implemented using C language. We use a well-known benchmark consisting of 20 VRPSPD instances to test our approach. This section presents results obtained by our ACS approach for those benchmark instances.

The problem set consists of 20 problem instances of 50 customers each one. These instances are divided into two different geographical scenarios: the scattered one namely SCA3-x and the concentrated one CON3-x. In the scattered scenario, a uniform distribution [0, 100] is used to randomly locate customer over a 100x100 geographical area. For the concentrated instances, again a uniform distribution [0, 100] is used to randomly locate customer half of the customer population, while a uniform distribution [100/3, 100/3] is used to randomly locate the other half of the customer population. Urban distributions where most of the customers are concentrated into 1/9th of the area are very similar to the concentrated scenario. Figures 1 and 2 show both scattered and concentrated scenarios respectively.

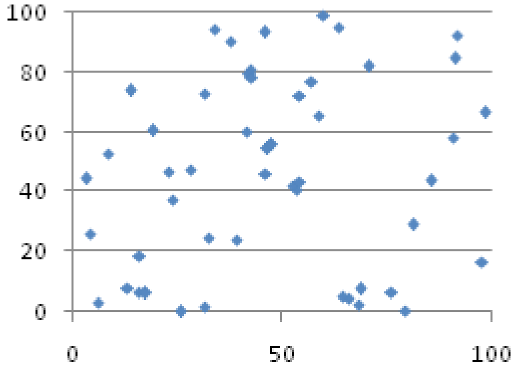

Figure 1. Customer randomly distributed over a 100x100 geographical area using a scattered distribution.

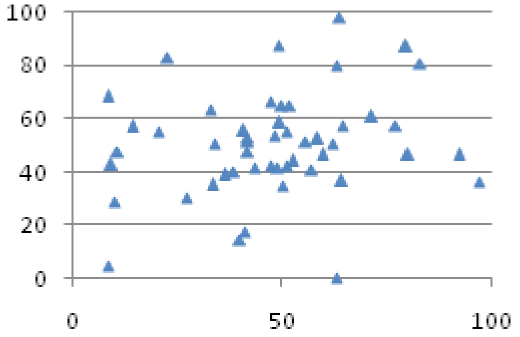

Figure 2. Customer randomly distributed over a 100x100 geographical area using a concentrated distribution.

Items to be delivered to the customers are uniformly distributed over the interval [0,100]. Items to be picked-up P_j corresponding to the delivery demand D_j is computed by using a random number r_j that is uniformly distributed over the interval [0, 1] such that $P_j=(0.5+r_j)D_j$ [2]. The transportation cost matrix is calculated as the Euclidean distance. The problem set we use to evaluate our ACS approach was introduced in [2] was also used in [4] and [13], where authors apply a TS strategy to solve the problem. Solution values obtained for each considered algorithm are summarized in Table 1. Our algorithm is run 20 times for each instance. Results showed here correspond to the average values obtained from those experiments.

As Table 1 shows, our ACS algorithm performs very well especially when compared to the algorithm presented in [2]. In that case, our algorithm reaches, in average, solution values that are 5,62% below results previously obtained in [2]. Moreover, when compared to the Tabu Search algorithm presented in [6], our algorithm is only better for some few instances. Despite our algorithm is, in average, 1% more expensive than the one presented in [6] it converges significantly faster which is also a desirable feature.

5. Conclusions and Future Work

The VRPSDP problem is a very important problem arising in the logistic industry. Unfortunately, this problem is quite complex from a mathematical point of view and obtaining optimal solutions for instances with a large number of decision variables using traditional mathematical programming techniques is not possible within a reasonable computational time. Thus, we need to explore other techniques different than the exact ones. In this context heuristic approaches allow us to obtain high quality solutions within an acceptable computational time. In this article we have considered an ACS algorithm to solve the problem.

Our ACS algorithm is shown to be quite competitive when compared to other techniques in literature. In this article, our ACS implementation outperforms results obtained previously in [2]. Furthermore, it is quite competitive when compared to other well-known approaches such as the one presented in [6].

Table 1. Obtained Results for all the 20 instances considered in this study.

Instance	RCRS[2]		TS-Heuristic		ACS	
	Best	Veh.	Best	Veh.	Best	Veh.
SCA3-0	689.0	-	640.6	4	656.4	4
SCA3-1	765.6	-	697.8	4	705.5	4
SCA3-2	742.8	-	659.3	4	662.2	4
SCA3-3	737.2	-	680.0	4	695.8	4
SCA3-4	747.1	-	690.5	4	712.2	4
SCA3-5	784.4	-	659.9	4	669.0	4
SCA3-6	720.4	-	653.8	4	669.9	4
SCA3-7	707.9	-	659.1	4	682.1	4
SCA3-8	807.2	-	719.5	4	733.3	4
SCA3-9	764.1	-	681.0	4	694.5	4
CON3-0	672.4	-	631.4	4	622.2	4
CON3-1	570.6	-	554.5	4	570.9	4
CON3-2	534.8	-	522.9	4	525.4	4
CON3-3	656.9	-	591.2	4	596.3	4
CON3-4	640.2	-	591.1	4	602.1	4
CON3-5	604.7	-	563.7	4	583.2	4
CON3-6	521.3	-	506.2	4	520.0	4
CON3-7	602.8	-	577.7	4	588.9	4
CON3-8	556.2	-	523.1	4	541.2	4
CON3-9	612.8	-	580.1	4	600.7	4

Moreover, our algorithm converges faster than other well-known local search strategies, which is also important particularly when very large scale problems are addressed.

As a future work, improved movements and rules could be implemented to improve the exploration features of our algorithm. Moreover, variants of the VRPSDP problem such as the one that incorporates time windows could be considered, this make the problem much more complex though.

REFERENCES

1. MIN, H., **The Multiple Vehicle Routing Problems with Simultaneous Delivery and Pick-up Points**. Transportation Research Part: A, vol. 23, 1989, pp. 377-386.
2. DETHLOFF, J., **Vehicle Routing and Reverse Logistics: The Vehicle Routing Problem with Simultaneous Delivery and Pick-up**. OR Spektrum, vol. 23, 2001, pp. 79-96.
3. NAGY, G., S. SALHI, **Heuristic Algorithms for Single and Multiple Depot Vehicle Routing Problems with Pickups and Deliveries**. European Journal of Operational Research, vol. 162(1), 2005, pp. 126-141.
4. MONTANÉ, T., R. GALVÃO, **A Tabu Search Algorithm for the Vehicle Routing Problem with Simultaneous Pick-up and Delivery Service**. Computers and Operations Research, vol. 33(3), 2006, pp. 595-619.
5. AI, J., KACHITVICHYANUKUL, V., **A Particle Swarm Optimization for the Vehicle Routing Problem with Simultaneous Pickup and Delivery**. Computers & Operations Research, vol. 36, 2009, pp. 1693-1702.
6. TANG, F., R. GALVÃO, **Vehicle Routing Problems with Simultaneous Pick-up and Delivery Service**. Journal of the Operational Research Society of India (OPSEARCH), vol. 39, 2002, pp. 19-33.
7. ZACHARIADIS, E., C. TARANTILIS, C. KIRANOUDIS, **A Hybrid Metaheuristic Algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pick-up Service**. Expert Systems with Applications, vol. 36, 2009, pp. 1070-1081.
8. BULLHEIMER, B., R. HARTL, C. STRAUSS, **An Improved Ant System Algorithm for the Vehicle Routing Problem**. Annals of Operations Research, vol. 89, 1999, pp. 319-328.
9. DORIGO, M., T. STÜTZLE, **The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances**, Handbook of Metaheuristics, International Series in Operations Research and Management Science, vol. 57, 2003, pp. 250-285.
10. DORIGO, M., L. GAMBARDELLA, **Ant Colonies for the Travelling Salesman Problem**. Biosystems vol. 43, 1997, pp. 273-81.
11. MANIEZZO, V., L. GAMBARDELLA, F. De LUIGI, **Ant Colony Optimization**, New Optimization Techniques in Engineering, by Onwubolu and Babu (Eds), Springer-Verlag Berlin Heidelberg, 2004, pp. 101-117.
12. DORIGO, M., L. GAMBARDELLA, **Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem**. IEEE Transactions on Evolutionary Computation vol. 1, 1996, pp. 53-66.
13. BIANCHESSI, N., G. RIGHINI, **Heuristic Algorithms for the Vehicle Routing Problem with Simultaneous Pick-up and Delivery**. Computers & Operations Research, vol. 34(2), 2007, pp. 578-594.
14. DING, Q., X. HU, L. SUN, Y. WANG, **An Improved Ant Colony Optimization and Its Application to Vehicle Routing Problem with Time Windows**. Neurocomputing, vol. 98(3), 2012, pp. 101-107.
15. GAJPAL, Y., P. ABAD, **An Ant Colony System (ACS) for Vehicle Routing Problem with Simultaneous Delivery and Pickup**. Computers & Operations Research, vol. 36, 2009, pp. 3215-3223.
16. JUN-MAN, K., Z. YI, **Application of an Improved Ant Colony Optimization on Generalized Traveling Salesman Problem**. Energy Procedia, vol. 17, 2012, pp. 319-325.
17. RIECK, J., J. ZIMMERMANN, **Exact Solutions to the Symmetric and Asymmetric Vehicle Routing Problem with Simultaneous Delivery and Pick-Up.** BuR - Business Research, vol. 6(1), 2013, pp. 77-92
18. ZHU, N., C. SHAO, **Vehicle Routing Problem with Simultaneous Delivery and Pick-up Based on the Improved Genetic Algorithm.** Fourth International Conference on Genetic and Evolutionary Computing, 2010, pp. 312-316

19. CAO, E., M. LAI, **An Improved Differential Evolution Algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pick-up Service**. Third International Conference on Natural Computation, 2007, pp. 436-440

20. BADR, A., **Solving Dynamic Vehicle Routing: An Alternative Metaheuristic Approach**, Studies in Informatics and Control, vol. 18 (2), 2009, pp. 159-164.

21. ÇATAY, B., **A New Saving based Ant Algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery.** Expert Systems with Applications, vol. 37 (10), 2010, pp. 6809-6817.