CAPACITATED VEHICLE ROUTING PROBLEM USING A NOVEL HYBRID SWARM OPTIMIZATION APPROACH

Jing Zhang[1], Jianguo Zheng[1]

ABSTRACT: This paper introduces a novel hybrid algorithm nature inspired approach based on particle swarm optimization, for successfully solving the capacitated vehicle routing problem. The hybrid particle swarm optimization incorporates a discrete particle swarm optimization algorithm, the multiple phase neighborhood search-greedy randomized adaptive search procedure algorithm, simulated annealing, and variable neighborhood search to enhance its search capabilities. The algorithm is suitable for solving very large-scale vehicle routing problems as well as other, more difficult combinatorial optimization problems. It is tested on a set of benchmark instances, simulation results and comparisons illustrate the effectiveness and efficiency of the algorithm.
KEY WORDS: optimization, discrete, simulated annealing, vehicle routing problem

INTRODUCTION

The capacitated vehicle routing problem (VRP) is well-known combinatorial optimization problem with considerable economic significance, which is a crucial issue in transportation and logistics systems. The problem can be described as the problem of designing routes from depots to a set of geographically scattered points (cities, warehouses, customers etc) with a fleet of vehicles. In recent years, The VRP has been largely studied because of the interest in its applications in logistic and supply-chains management.

During the last decade nature inspired intelligence became increasingly popular through the development and utilization of intelligent paradigms in advanced information systems design. These methods contribute to technological advances driven by concepts from nature/biology including advances in structural genomics, mapping of genes to proteins and proteins to genes, modeling of complete cell structures, functional genomics, self-organization of natural systems, etc. because of the recent advancements in evolutionary computation, In the literature, many researchers have proposed much new evolutionary algorithm and made significant efforts to solve the VRP in supply chain management and logistics.

Ai and Kachitvichyanukul (2009) developed a PSO for a VRP with simultaneous pick-up and delivery, and compared the performance of their method with other existing metaheuristics using some benchmark problems. They used a similar PSO for the capacitated VRP (CVRP) and reported some promising results (Ai & Kachitvichyanukul, 2009). Babak Farhang Moghaddam (2012) presented a new decoding algorithm and applying PSO to solve the capacitated VRP. ACO has successfully been applied to solve capacitated vehicle routing problems [Yao & Yao, 2007]. Recently, Alvarenga et al (2007) combined the GA with a set partition formulation, Ghoseiri and Ghannadpour (2009) used a push-forward insertion heuristic in the initialization of the GA. Nazif and Lee (2010) designed a specific crossover operator that is based on graph theory for the VRP with time windows. Wang and Lu (2009) propose a novel hybrid genetic algorithm (HGA) for solving a capacitated vehicle routing problem (CVRP). Two real cases one for locally active distribution and another for arms part transportation at a combined maintenance facility, both involving the Taiwanese armed forces are used to detail the analytical process and demonstrate the practicability of the proposed HGA for optimizing the CVRP.

In the residential waste collection problem, vehicles move along streets to collect garbage accumulated from residential areas. Therefore, the routing problem can be considered as an arc routing problem (Bonomo et al, 2013), Wy et al (2013) discussed various real-world issues in Rollon–rolloff vehicle routing problem, such as time windows, changing service types, and heterogeneous vehicles, and proposed an iterative heuristic approach. Goksal et al (2013) consider vehicle routing problem with simultaneous pickup and delivery.

Particle swarm optimization (PSO) is inspired by social behavior simulation, was originally designed

[1] Glorious Sun School of Business and Management DongHua University, Shanghai, China

and developed (Eberhart and Kennedy, 1995). The original PSO is used for continuous optimizations (Wu and Zheng, 2012). It is a population-based search algorithm that was on the basis of the simulation of the social behavior of birds within a flock. Most applications of PSO have concentrated on the optimization in continuous space while some work has been done to the discrete optimization. The PSO is a very popular optimization method and its wide use, mainly during the last years, is due to the number of advantages that this method has, compared to other optimization methods.

In this paper, we proposed a hybrid metaheuristic optimization method, called novel hybrid discrete swarm intelligence optimization (HDPSO-CVRP) for solving the capacitated vehicle routing problem (CVRP), which allows the definition of scheduling algorithms by appropriately selecting and combining several different features of discrete particle swarm optimization, simulated annealing, and variable neighborhood search. The rest of this paper is organized as follows. Section 2 briefly describes the CVRP and the mathematical formulation. The section 3 illustrates the detailed implementation steps of HDPSO-CVRP algorithm to solve the CVRP. Section 4 compares our experimental results with the recent algorithms that have been used to solve the CVRP. Finally, the conclusions are discussed in Section 5.

CAPACITATED VEHICLE ROUTING PROBLEM

The capacitated vehicle routing problem is often described as the problem in which vehicles based on a central depot are required to visit geographically dispersed customers in order to fulfill known customer demands. It can be defined as follows: n customers are waiting to be served, each of which requires a quantity qi of goods (i = 1, 2. . . n). A depot has a fleet to deliver the goods. Each vehicle has a capacity of Q units, i.e., the total demand of customers served on each route could not exceed Q. Therefore, the vehicle has to periodically return to the depot for reloading, which is equivalent to dispatching a new vehicle for delivery. Besides, each customer must be visited once and only once by exactly one vehicle. A solution of the CVRP is a collection of routes that a vehicle starts from the depot, serves some specific customers, and terminates at the depot.

Let G=(V,E) be a directed complete graph, where V={n0, n1, n2,…nn} is the node(customer) set, (n0 refers to the depot and the customers are indexed n1, n2, ... nn) and E={(nl, nm): n1, n2 V, nl≠nm } is the edge set. Each customer vertex is associated with a demand quantity qi, among which node n0 is associated with q0=0.each edge ((nl, nm)) is associated with a travel time tij, which is represented by the Euclidean distance di, j between nodes nl and nm (dlm=dml). The objective of the CVRP is to minimize the total travel time of all the vehicles. ti is the arrival time at customer i; wi is the wait time at customer i; K is the total number of vehicles; N is the depot and customers set; cij is the length on the link from customer i to j; tij is travel time between customer i and j; qi is the quantity of the goods to be delivered at customer i; Q is the capacity of a vehicle. The objective is to minimize the total travel length under several constraints: (2) the maximum number of routes constraint,(3) travel constraints, (4 and 5) service constraints, (6) the capacity constraint.

$$\min \sum_{i=0}^{N} \sum_{j=0, j\neq i}^{N} \sum_{K=1}^{K} c_{ij} x_{ijk} \quad (1)$$

$$s.t.$$

$$\sum_{i=0}^{K} \sum_{j=1}^{N} x_{ijk} \le K \qquad for\ \ i = 0 \quad (2)$$

$$\sum_{j=1}^{N} x_{ijk} = \sum_{j=1}^{N} x_{jik} \le 1 \qquad for\ \ i = 0\ \ k \in \{1, ..., K\} \quad (3)$$

$$\sum_{k=1}^{N} \sum_{j=0, j\neq i}^{N} x_{ijk} = 1 \qquad for\ \ i \in \{1, ..., N\} \quad (4)$$

$$\sum_{k=1}^{K} \sum_{i=0, i\neq j}^{N} x_{ijk} = 1 \qquad for\ \ j \in \{1, ..., N\} \quad (5)$$

$$\sum_{k=1}^{N} q_i \sum_{j=0, j\neq i}^{N} x_{ijk} \le Q \qquad for\ \ k \in \{1, ..., K\} \quad (6)$$

$$where \ \ x_{ijk} = \begin{cases} 1 & \text{the link from customer node i to j is visited by vehicle k} \\ 0 & \text{otherwise} \end{cases}$$

THE PROPOSED HDPSO-CVRP ALGORITHM

3.1 General recommendations

Most of the population based stochastic search algorithms for optimization problems work on the basis of two contradictory aspects: exploration and exploitation. Exploration refers to ability of the algorithm to ‘explore’ or search different regions of the feasible search space whereas ‘exploitation’ means the ability of convergence of all the particles to the near optimal solutions as fast as possible. Due to excessive exploitative tendency, in many cases, the population may lose its global exploration abilities within a relatively small number of generations, thereafter leading to premature convergence. In this context, we propose a modified scheme on the learning strategy by introducing neighbourhood based selection of the exemplar for velocity updating. In the hybrid HDPSO-CVRP algorithm, each particle moves through space while updating its own best position, the global best position. It informs other particles about its best position, it also obtains theirs and then adjusts its own position and velocity according to the shared information. The particles have a tendency to fly towards better search regions over the course of search process. The velocity and position updates of the dth dimension of the ith particle are presented below Eq. (7) and Eq. (8):

$$v_i^d(t+1) = w \times v_i^d(t) + c_1 \times rand_1^d \times \left(pbest_{fi}^d(t) - x_{id}(t)\right) + c_2 \times rand_2^d \times \left(gbest^d(t) - x_i^d(t)\right) \quad (7)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t) \quad (8)$$

In Eq.(1),w is the inertia weight used to balance between the global and local search abilities.

$$w = w_{\max} - (w_{\max} - w_{\min}) \times t / t_{\max} \quad (9)$$

$w_{\max}$ is the initial inertia weight of the velocity, $w_{\min}$ is the final inertia weight of the velocity. I is the current iteration times,

$I_{\max}$ is the total iteration times. The HDPSO-CVRP algorithm is presented in figure 1.

Algorithm 1:The HDPSO-CVRP algorithm

Initialize
Initialize the population size
Generate the initial population of the particles using MPNS–GRASP
Evaluate the fitness function of each particle
Keep the best global position of the whole swarm
Keep the best local position of each particle Main Phase
Main Phase
Do a loop (the maximum number of generations reached)
Calculate the velocity of each particle using Eq.(7) and Eq.(9)
Calculate the new position of each particle using Eq.(8)
Evaluate the new fitness function of each particle
Apply Variable neighborhood search
Update the best local position (pbest)of each particle
Update the best global position (gbest)of the whole swarm
Local improvement from gbest
{Generate Initial solution S*
Set initial temperature To and final temperature Tend;
While current temperature> Tend do
Apply Variable neighbourhood search to find a solution Sn
If f(S*)<= f(Sn)then set S*=Sn
Else accept Sn as new solution with probability p(To, S*,Sn) using Eq.(10)
End if
Adapt temperature(To)
End while
}
Update the best global position (gbest)of the whole swarm
Update the inertia weight
End do
Return the particle with the best global position of the whole swarm (the best solution).

Fig 1. Pseudo code of HDPSO-CVRP

As we know, the problem space of the VRPTW is a directed complete graph, i.e., G = (V, E), and each candidate solution is a spanning subgraph of G. The candidate solutions are judged to be feasible solutions if they satisfy the constraints that are defined by formulas (2)–(6). The goal of the CVRP is to find a feasible solution that minimizes the total travel distance (TD).

In the HDPSO-CVRP, the search space of the particle swarm is the arc set E of the complete graph G. Each particle's position is represented by a crisp set of arcs, which is a subset of A, to denote a feasible solution of the problem, the constraints-based decoder in the HDPSO-CVRP was shown in figure 2. The position and velocity in the HDPSO-CVRP are represented the nodes of arc sets as follows.

One of the key issues in designing a successful PSO for vehicle routing problem is to find a suitable mapping between vehicle routing problem solutions and particles in PSO. Each particle is recorded via the path representation of the tour, that is, via the specific sequence of the nodes. The position of each individual (called particle) is represented by a d-dimensional vector in problem space, i=1, 2… N(N is the population size),and its performance is evaluated on the predefined fitness function. Concerning the fitness function, it should be noted that in VRP, the fitness of each particle is related to the route length of each circle and since the problem that we deal with is a minimization problem, if a feasible solution has a high objective function value then it is characterized as an unpromising solution candidate.

Initially all the solutions (particles) are represented with the path representation of the tour.

As the calculation of the velocity of each particle is performed by the Eq.(1), the above mentioned representation should be transformed appropriately. We transform each element of the solution into a floating point interval [0, 1], calculate the velocities of all particles and then convert back into the integer domain using relative position indexing. For example if the velocities' vector of a particle is 0.3 1 0.7 0.9 0.5, the backward transformation gives 1 5 3 4 2.

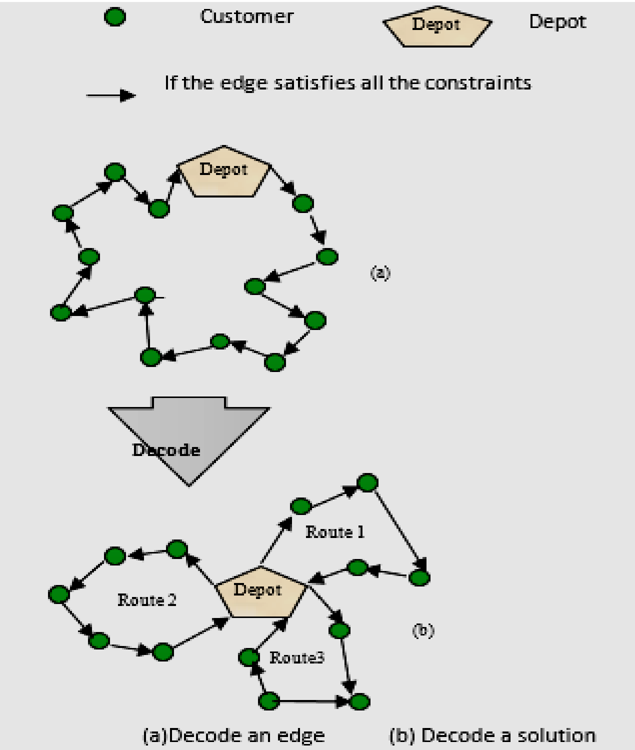

Fig 2. Illustrations of the constraints-based decoder in the HDPSO-CVRP

(2) Initial population

The multiple phase neighborhood search (MPNS–GRASP) is used to initialize the population. Instead of using randomly generated initial populations which may or may not necessarily contain good candidate solutions, MPNS–GRASP introduces the flexibility of applying alternative greedy functions in each iteration instead of only one simple greedy function as in the classical approach. Moreover, a combination of greedy functions is also possible. The algorithm starts with one greedy function and if the results are not improving, an alternative greedy function is used instead. In these greedy functions, initially a travelling salesman problem is solved (Marinakis et al, 2005), disregarding the side constraints (capacity constraints and maximum route duration constraints) of the vehicle routing problem. Subsequently, the solution of the TSP is converted to a solution of the VRP by adding the side constraints. More precisely, the first vehicle route begins from the node that corresponds to the depot and moves to the next node (customer) based on the solution of the TSP, checking if the capacity of the vehicle or if the maximum route length of the vehicle are not violated. If any of these two constraints are violated, then the vehicle returns to the depot and a new route begins.

3.2 Variable neighbourhood search (VNS)

VNS is one of the very well-known local search methods. Large numbers of papers have attempted to improve upon solutions by using a relatively large arsenal of local search improvement heuristics, based around different neighbors. This is simply because of its ease of use and success in solving combinatorial optimization problems. In the following, a specific local search for each neighborhood is applied. The local searches are listed as figure 3.

Algorithm 1:The Local searches
For Ll(s): Step 1: Choose randomly a routes R. Step 2: Choose randomly two vehicles N1 and N2 on route R. Step 3: Swap vehicle N1 and N2. For L2(s): Step 1: Choose randomly two routes R1 and R2. Step 2: Choose randomly a vehicles N1 on R1and a vehicle N2 on route R2. Step 3: Swap vehicle N1 and N2. For L3(s): Step 1: Choose the route R that has the fewest loading capacity. Step 2: For each customer in R, try to insert it into the other vehicle routes on the premise such that the insertion satisfies the constraints of the CVRP. Step 3: If all the customers in R can be inserted into other routes, route R is deleted.

Fig 3. Pseudo code of local searches

4 NUMERICAL RESULTS

In the experiments, the inertia weight ωin Eq. (9) is initialized as 0.9 and linearly decreased from 0.9 to 0.4 during the training process, and the acceleration coefficient c1=c2 is set as 2.0. The population size is set as M = 20. The parameters of HDPSO-CVRP algorithms are selected after thorough testing. Our algorithm is coded in matlab, each test is conducted 30 times independently. The algorithms were tested on a set of benchmark problems, the 14 benchmark problems proposed (Christofides et al. 1979) in the table 1.

Table 1. Manufacturing vs. Remanufacturing

	Nodes	Capacity	Max.tour length	Optimal solution
1	51	160	∞	524.61
2	76	140	∞	835.26
3	101	200	∞	826.14
4	151	200	∞	1028.42
5	200	200	∞	1291.45
6	51	160	200	555.43
7	76	140	160	909.68
8	101	200	230	865.94
9	151	200	200	1162.55
10	200	200	200	1395.85
11	121	200	∞	1042.11
12	101	200	∞	819.56
13	121	200	720	1541.14
14	101	200	1040	866.37

Each instance of the set contains between 51 and 200 nodes including the depot. The efficiency of the HDPSO-CVRP algorithm is measured by the quality of the produced solutions. The error is given in terms of the relative deviation from the best known solution, that is

$$error = \frac{|total\ cost\ of\ a\ heuristic\ solution - optimal\ solution|}{total\ cost\ of\ a\ heuristic\ solution} * 100\% \qquad (10)$$

To verify the effectiveness and efficiency of our proposed algorithm, in Table V the cost and the computational time of all implementations by four algorithms such as HDPSO-CVRP, PSO, D-Ants (Reimann et al, 2004), Tabu search (Lee et al, 2006) are presented in table 2. From this table, it can be observed that the use each of the characteristics of the HDPSO-CVRP improve significantly either the quality of the solution or the computational time or both of them. All algorithms were run 30 times for each instance and the results presented include the best solution and the computational time, worst and average solutions found.

Table 2. Results of four algorithm benchmark instances

	HDPSO-CVRP			HDPSO-CVRP		
	best	error%	Cpu	Cost	error%	Cpu
1	524.61	0.00	0.06	524.61	0.00	1.09
2	835.26	0.00	0.42	845.24	1.18	1.75
3	827.14	0.12	0.82	832.91	0.81	1.84
4	1028.4	0.00	1.09	1049.24	1.98	2.06
5	1294.4	0.23	2.35	1361.24	5.13	3.67
6	555.43	0.00	0.45	555.43	0.00	1.19
7	909.68	0.00	0.48	918.27	0.94	1.94
8	865.94	0.00	1.29	889.12	2.61	2.13
9	1162.5	0.00	1.27	1197.23	2.90	2.35
10	1398.8	0.21	3.01	1428.21	2.27	3.98
11	1045.1	0.29	1.38	1057.01	1.41	1.69
12	819.56	0.00	1.64	827.21	0.92	2.07
13	154	0.3	1.59	1551.21	0.65	1.73
14	866.37	0.00	1.76	876.45	1.15	2.01
Average error		0.08	1.26		1.57	2.11

	PSO			D-Ants		
	best	error%	Cpu	best	error%	Cpu
1	524.61	0.00	0.96	524.61	0.00	1.21
2	835.26	0.00	1.39	835.26	0.00	1.15
3	828.96	0.34	1.56	830.24	0.49	1.24
4	1028.42	0.00	1.63	1028.42	0.00	1.84
5	1308.24	1.28	2.69	1318.04	2.02	3.04
6	555.43	0.00	0.74	555.43	0.00	1.09
7	912.34	0.29	1.56	916.76	0.77	1.45
8	865.94	0.00	1.64	865.94	0.00	1.86
9	1162.55	0.00	1.65	1168.93	0.55	1.53
10	1406.47	0.76	3.24	1415.48	1.39	3.24
11	1053.54	1.08	1.42	1054.03	1.13	1.57
12	825.38	0.71	1.87	819.56	0.00	1.97
13	1556.87	1.01	1.76	1559.72	1.19	1.85
14	874.58	0.94	1.80	868.37	0.23	1.92
Average error		0.4	1.7		0.5	1.7

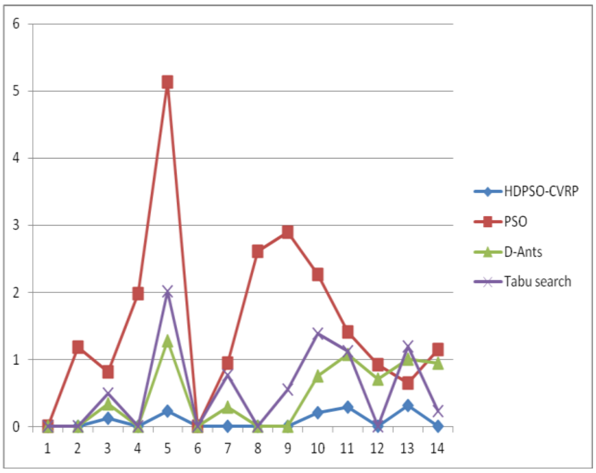

Fig 2. Comparison of the proposed algorithm with other algorithm based on the error

In table 2, the results of the HDPSO-CVRP algorithm are also compared with the results of a number of implementations of the PSO, D-Ants and Tabu search. In these implementations, the same instances are used as in this paper and comparisons of the results can be performed. As it can be observed that the values of Best, Average of the HDPSO-CVRP algorithm perform better than other three algorithms for all experiments. We can also see that the ability to improve the values of average and error of the DMPSO-ACO algorithm is the best among the three algorithms. Comparison of the proposed algorithm with other algorithm based on the quality and the computational time as shown in figure 2.

5 CONCLUSION

In this paper, we have presented a novel hybrid swarm intelligence optimization algorithm named HDPSO-CVRP algorithm for the capacitated vehicle routing problem. In the HDPSO-CVRP, We have developed a novel decoding method for interpreting PSO solutions for the CVRP, simulated annealing and three local search operators in a VNS loop in order to significantly improve the quality of the solutions. The algorithm was applied in a set of benchmark instances and gave very satisfactory results. From the experimental results, we can see that the best solution quality of HDPSO-CVRP algorithm is really better than those of the PSO, D-Ants and hybrid Tabu search respectively. In addition, the DMPSO-ACO is simple and easy to implement.

6 ACKNOWLEDGEMENTS

This research was supported by the Natural Science Foundation of Shanghai (15ZR1401600).

7 REFERENCES

► [1] Ai, J., & Kachitvichyanukul, V. (2009a). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. Computers &Operations Research, 36(5): 1693-1702.

► [2] Ai, J., & Kachitvichyanukul, V. (2009b). Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. Computers & Industrial Engineering, 56(1): 380-387.

► [3] Babak Farhang Moghaddam, Rubén Ruiz. (2012). Seyed Jafar Sadjadi, Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm, Computers & Industrial Engineering, 62(1): 306-317.

► [4] Yao, J. B., & Yao, B. Z. (2007). A parallel ant colony optimization for multi-depot vehicle routing problem. In: Proceedings of the 2007 international conference on computational intelligence and security workshops table of contents, 240-243.

► [5] G. B. Alvarenga, G. R. Mateus, and G. D. Tomi, (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows, Comput. Oper. Res, 34(6): 1561-1584

► [6] K. Ghoseiri and S. F. Ghannadpour. (2009). Hybrid genetic algorithm for vehicle routing and scheduling problem, J. Appl. Sci., 9(1): 79-87.

► [7] H. Nazif and L. S. Lee. (2010). Optimized crossover genetic algorithm for vehicle routing problem with time windows, Amer. J. Appl. Sci., 7(1): 95-101.

► [8] Wang, C. H., & Lu, J. Z. (2009). An effective evolutionary algorithm for the practical capacitated vehicle routing problems. Journal of Intelligent Manufacturing, doi:10.1007/s10845-008-0185-2.

► [9] Bonomo F, Duran G, Larumbe F, MarencoJ. (2013). A method for optimizing waste collection using mathematical programming: a Buenos Aires case study. Waste Management & Research, 30(3): 311-324.

► [10] Wy J, Kim B, Kim S. (2013). The rollon–rolloff waste collection vehicle routing problem with real-world issues. European Journal of Operational Research, 224(3): 466-76.

► [11] V. Savithri, Purushothaman. (2013). Intima-media thickness estimation by back propagation algorithm, Int. J. of Convergence Computing, 1(1): 3-18

► [12] J. Kennedy, R.C. Eberhart. (1995). Particle swarm optimization, Proceeding of the IEEE International Conference on Neural Networks 4, 1942-1948.

► [13] Daqing Wu, Jianguo Zheng. (2012). “A Dynamic Multistage Hybrid Swarm Intelligence Optimization Algorithm for Function Optimization,” Discrete Dynamics in Nature and Society, vol. 2012, Article ID 578064, 22 pages.

► [14] Marinakis, Y., Migdalas, A., Pardalos, P.M.(2005a). Expanding neighborhood GRASP for the traveling salesman problem. Computational Optimization and Applications, 32(3), 231-257.

► [15] Christofides, N., Mingozzi, A., Toth, P. (1979). The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (Eds.), Combinatorial Optimization. Wiley, Chichester.

► [16] Reimann, M., Doerner, K., Hartl, R.F. (2004). D-Ants: savings based ants divide and conquer the vehicle routing problem. Computers and Operations Research 31 (4), 563-591.

► [17] Lee, Y. H., Jung, W. J., & Lee, K. M (2006). Vehicle routing scheduling for cross-docking in the supply chain. Computers &Industrial Engineering, 51(2), 247-2