A Heuristic Based on Integer Programming for the Vehicle Routing Problem with Backhauls

Gulcin Dinc Yalcin and Nihal Erginel
Department of Industrial Engineering
Anadolu University, Eskisehir, Turkey

Abstract

Vehicle Routing Problem with Backhauls (VRPB) is an extension of vehicle routing problem. In this problem, customers are divided into two sets as linehaul and backhaul customers. All goods are delivered to linehaul customers before goods are brought from backhaul customers. In this study, a heuristic based on integer programming is proposed. Firstly linehaul and backhaul customers are clustered to vehicles with the proposed integer model. Then clusters are routed as Travelling Salesmen Problem with Backhaul (TSPB) with proposed TSPB integer model. Finally the route is improved by local search. The proposed heuristic is the first of heuristics for VRPB because it applies only integer programming model in clustering and routing phases. Also a set of benchmark problem from literature is solved by the proposed heuristic and the results are given and discussed.

Keywords
Vehicle routing problem, backhauls, integer programming, heuristic.

1. Introduction
The Vehicle Routing Problem (VRP) is to deliver goods to a set of customers with the purpose of minimizing total delivery distance or minimizing total delivery time under the constraint of vehicle capacity and customer demand. A customer is served by one vehicle which starts from depot with loaded and returns to depot after delivering goods to customers. Danzig and Ramser [11] introduced this problem in 1959. So many approaches which can be divided into heuristic approaches and metaheuristic approaches are proposed in literature by this time.

Heuristics approaches can be classified into constructive heuristics, two phase heuristics, and improvement heuristics [17] Saving algorithm [10] is the most known of constructive heuristic. Two phase heuristics are divided as cluster- first route-second methods that is firstly customers are assigned to vehicles then vehicles are routed, and route-first cluster-second methods that is firstly routes are generated as Travelling Salesman Problem (TSP) then the route is divided into feasible vehicles routes. Fisher and Jaikumar Algorithm [12] that is based on generalized assignment problem, Baker and Sheasby Algorithm [4] that is an extension of the Fisher and Jaikumar Algorithm and Sweep algorithm [17] are cluster-first route-second methods. Dijkstra [13]'s algorithm is a route-first, cluster-second method. Improvement heuristics are based on Lin's λ-opt approach [22] .

Metaheuristic approaches such as simulated annealing, genetic algorithms, tabu search, ant colonies, particle swarm optimization and the hybrid of them are applied to VRP. Alfa et. al [2] are used simulate annealing that combined with 3-opt heuristic. Van [41] is proposed simulated annealing based improvement methods. Osman [31] developed tabu search algorithm and a hybrid of simulated annealing and tabu search to solve VRP. TABUROUTE algorithm is a tabu search metaheuristics for VRP that is developed based on generalized insertion procedure by Gendreau et al. [16]. Xu et. al. [43] proposed a tabu search based on network flow local search approach. Barbarosoglu et. al. [5] developed a new tabu search algorithm that proposes a new neighborhood generation procedure and are not included the classical diversification step. A new variant of tabu search that is named as granular tabu search is used for VRP by Toth et. al. [36]. Cardeau and Maischberger [9] used tabu search with iterated local search for VRP and some variant of VRP (the periodic VRP, mılti-depot VRP, site-depented VRP). Bullnheimer et. al. [8] used ant system algorithm, Mazzeo et. al. [28] used ant colony algorithm, Raimann et. al. [32] used saving based ant system, Bin et. al. [6] used ant colony optimization that is combined with the mutation operation of genetic algorithm, Zhang et. al. [44] used ant colony optimization with scatter search to solve VRP. Lee et. al. [23] proposed an enhanced ant colony optimization that is a hybrid of simulated annealing and ant colony optimization. Baker et. al. [4] used genetic algorithm for VRP, Nazif And Lee [30] used genetic algorithm with optimized crossover operator, Lin et. al. [24] used a hybrid algorithm of simulated annealing and tabu search, Ai et. al. [1] used particle swarm optimization, Marinakis et. al. [27] used a hybrid of genetic and particle swarm optimization algorithm, Marikanis et. al. [26] used hybrid particle swarm optimization algorithm that combines a particle swarm optimization and the multiple phase neighborhood search-greedy randomized adaptive search procedure. Goetha, Poonthalir and Vanathi [18] used partical swarm optimization with crossover and mutation operators for better exploration of particles.

There are different variants of VRP. One of them is Vehicle Routing Problem with Backhauls (VRPB) where customers are grouped as linehaul customers and backhaul customers. Products are delivered to linehaul customers and products are picked up from backhaul customers. In this problem, a customer is served one vehicle that starts and finishes the route at depot under the vehicle capacity constraint and backhaul customers can be visited by a vehicle after the vehicle finishes linehaul customers. Some mathematical formulations of VRPB are defined by Goetschalckx and Jacobs-Blecha [19], Toth et. al. [38], Mingozzi et. al [29]. Solution procedures can be classified as exact algorithms, heuristic approaches and metaheuristic approaches. Exact algorithms that used branch and bound algorithm are developed by Toth and Vigo [38] and Mingozzi et. al. [29]. Toth and Vigo [38] used Lagrangian relaxian algorithm and Mingozzi et. al. [29] used a heuristic to find lower bound.

Deif and Bodin [12] proposed an extension of saving algorithm. Goetschalckx and Jacobs-Blecha [19] proposed a two phase solution methodology that the first phase is based on spacefilling curves. Also Goetschalckx Jacobs-Blecha [20] presented a heuristic that an extension of Fisher and Jaikumar's [14] the generalized assignment heuristic for VRP. Salhi et. al. [34] investigated an extension to the classical insertion-based heuristic for single and multi-depots. Toth and Vigo [37] proposed a cluster-fist route-second heuristic for both symmetric and asymmetric cost matrix. Ropke and Pisinger [33] proposed a unified heuristic for different variants of VRPB. Tutuncu et. al. [40] developed a visual interactive approach that is based on Greedy Randomized Adaptive Memory Programing Search and developed a decision support system for VRPB and mixed VRPB. Wang and Wang [42] proposed a novel two phase heuristic that the initial routes are generated by using modified parallel C-W saving heuristic in the first phase and the initial solution is improved by reactive tabu search in the second phase.

Tavakkaoli-Moghaddam et. al. [35] developed a memetic algorithm with different local search algorithms and initial solutions are generated by using greedy heuristic. Brandão [7] presented a tabu search algorithm that used pseudo-lower bounds for initial solution. Liu, Li and Wu [25] propsed tabu search to minimize the number of vehicle and total travel distance. Gajpal and Abad [15] used multi ant colony system that the first type of ant is used to assign customers to vehicles and the second type of ant is used to construct a route for each vehicle. Zachariadis and Kiranoudis [44] is proposed a local search metaheuristic that uses exchanging variable-length customer sequences with Static Move Description.

In this study, we proposed a heuristic based on integer programming to solve VRPB. Our heuristic is similar to cluster-first route-second heuristics for VRP and it is the first heuristic that use only integer programming for clustering and routing phases. In the clustering phase, customers are assigned to vehicles by proposed integer model that define the seed customers equal to number of vehicles and assign other customers to seed customers with minimizing total distance within clusters. In the route phase, each cluster is routed as Travelling Salesman with Backhaul (TSPB) by TSPB integer model. The model is a modification of Goetschalckx and Jacobs-Blecha's [19] model for VRPB with omitting capacity and assigning constraints. After solving two phase, solution is improved by local search. A set of benchmark problem from literature is solved.

The remaining part of the study is organized as follows. In Section 2, the proposed heuristic is described in details for two phases and local search. The results of benchmark problems that are generated by the proposed heuristic are given and discussed in Section 3. Conclusions follow in Section 4.

2. A Heuristic Based on Integer Programming

In this study, a heuristic based on integer programming is proposed for VRPB that is an extension of VRP. The proposed heuristic has two phases that is based on cluster-first, route-second and local search that uses insertion operation and interchange operation to improve the routes.

Phase 1: An integer programming model is proposed for clustering customers to vehicles. In this phase, the proposed integer program defines the seed customers as much as number of vehicles and then assigns other customers to them.

Phase 2: Each cluster is routed as TSPB with integer programming model because the customers are divided into clusters and the number of customers in clusters is small enough to use integer programming. The integer programming model is modified from Goetschalckx and Jacobs-Blecha's [19] model for VRPB by removing constraints that are about vehicle capacity and assigning customers to vehicles.

Local Search: Local search with insertion and interchange operations are applied to improve the routes in other word to reduce total travel distance. Insertion operation tries to take a customer from its original vehicle and put it in another vehicle if the total travel distance is reduced. Interchange operation tries to change two customers from different vehicles and put them the best position in their new vehicles if the total travel distance is reduced.

In following sections, initially first phase that is named as the clustering phase, then second phase that is named as the routing phase is explained and integer programming formulation of the two phases is given. Then local search for using to improve the route is described.

2.1. Phase 1: Clustering Customers to Vehicles

A linehaul customer or a backhaul customer is served only one vehicle. The clustering phase assign a customer to a vehicle and the customers that is assigned the vehicle is tried to be close each other as possible. The proposed integer programming model is used for this phase. The model defines the seed customers for each cluster that is equal to the number of vehicles. Other customers are assigned to a seed customer, in a sense to a vehicle, due to minimize distance between the customers and seed customers. The proposed integer programming model for the cluster phase is as:

<u>Sets</u>

i, j linehaul and backhaul customer nodes

<u>Constants</u>

N	number of linehaul customers	b_i	demand of backhaul customer (i=N+1, …, N+M)
M	number of backhaul customers	C	capacity of vehicles
d_{ij}	distance between node i and node j	K	number of vehicles
a_i	demand of linehaul customer (i=1,…, N)		

<u>Variables</u>

$$x_{ij} = \begin{cases} 1 & \text{if node i is assigned to node j} \\ 0 & \text{otherwise} \end{cases}$$

<u>Formulation</u>

$$\text{minimize } z_1 = \sum_i \sum_j d_{ij} x_{ij} \tag{1}$$

Subject to:

$$\sum_{j=1}^{N+M} x_{ij} = 1 \qquad \forall i = 1, .., N + M \tag{2}$$

$$\sum_{i=1}^{N} a_i x_{ij} \leq C x_{jj} \qquad \forall j = 1, .., N + M \tag{3}$$

$$\sum_{i=N+1}^{N+M} b_i x_{ij} \leq Cx_{jj} \quad \forall j = 1,..,N+M \tag{4}$$

$$\sum_{i=1}^{N+M} \sum_{\substack{j=1 \\ i=j}}^{N+M} x_{ij} = K \tag{5}$$

$$x_{ij} \in \{0,1\} \tag{6}$$

The objective of the model (1) is to minimize the total distance that is the sum of distance between a customer and the relevant seed customer in each cluster. Constraint (2) represents that each customer must be assign only one cluster. Constraints (3) and (4) say that total demand of linehaul customers and backhaul customers that is assigned to a cluster cannot be exceed capacity of vehicle and if a customer is seed customer than other customers can be assigned. Constraint (5) indicates that the number of clusters is equal to the number of vehicle.

2.2. Phase 2: Routing Each Clusters

After the customers are clustered, the clusters can be routed as TSPB. The number of customers in clusters is small enough to solve the problem with integer programming. So an integer programming model is proposed by modification of Goetschalckx and Jacobs-Blecha's [19] model for VRPB. The modification is to omit the constraints about vehicle capacity and assigning customers to vehicles. The modified integer programming model is used for the routing phase and is solved for each cluster. The proposed integer programming model for the route phase is as:

Sets

i, j linehaul and backhaul customer nodes of relevant cluster and depot

Constants

n number of linehaul customers for relevant cluster

m number of backhaul customers for relevant cluster

c_{ij} distance between node i and node j

Variables

$$y_{ij} = \begin{cases} 1 & \text{if vehicle travels from node i to node j} \\ 0 & \text{otherwise} \end{cases}$$

Formulation

$$\text{minimize } z_1 = \sum_i \sum_j c_{ij} y_{ij} \tag{7}$$

Subject to:

$$\sum_{i=0}^{n} y_{ij} = 1 \quad \forall j = 1,..,n \tag{8}$$

$$\sum_{j=0}^{n+m} y_{ij} = 1 \quad \forall i = 0,..,n \tag{9}$$

$$\sum_{i=0}^{n+m} y_{ij} = 1 \quad \forall j = n+1,..,n+m \tag{10}$$

$$\sum_{\substack{J=n+1 \\ j=0}}^{n+m} y_{ij} = 1 \quad \forall i = n+1,..,n+m \tag{11}$$

$$\sum_{i=0}^{n} \sum_{j=n+1}^{n+m} y_{ij} \leq 1 \tag{12}$$

$$\alpha_i - \alpha_j + (n+m+1)\, y_{ij} \leq (n+m) \quad \forall i,j = 1,..,n+m \;\; and \;\; i \neq j \tag{13}$$

$$y_{ij} \in \{0,1\} \tag{14}$$

The objective of the model (7) is to minimize the route distance. Constraints (8) and (9) indicate that there can be only one passing between a linehaul customer/the depot and a linehaul customer. Constraints (10) says that there can be only one passing for a backhaul customer from a linehaul customer or a backhaul customer or the depot and constraints (11) says that after a backhaul customer, the passing can be made to a backhaul customer or the depot. Constraint (12) represents that there should be at most one link between linehaul customers and backhaul customers. Constraint (14) is the subtour elimination constraint of TSP.

2.3. Local Search

The solution that is obtained after the two phases is improved by local search. Insertion and interchange operations are applied to improve. In insertion operation, a customer is removed from its original vehicle and inserted in another vehicle with checking capacity of inserting vehicle. If total distance is better after inserting the customer in new vehicle, new route is obtained. There are some rules for insertion to provide VRPB conditions. A linehaul customer can be inserted after the depot or another linehaul customer. A backhaul customer can be inserted after a backhaul customer or between a linehaul customer and backhaul customer or before the depot. In interchange operation, two customers that are in different vehicle are exchanged with checking the capacity of vehicles. Customers are positioned the best place in their new vehicles if total distance is reduced. Rules for interchange operations are the same as insertion operation.

3. Computational Experiments

In this section, values of solutions that are obtained by proposed heuristic are presented. Goeschalckx and Jacobs's benchmark problem set are solved as other studies in literature. In this problems, customers x coordinates are uniform distributed in the interval [0,2400] and customer y coordinates are uniform distributed in the interval [0,32000]. The coordinate of the depot is (12000,16000) for all problems and the depot is located centrally. This problem set consist of 14 main problem and 62 sub problems that are obtained from 14 main problem by different vehicle capacity and number of vehicle.

Gams with Cplex 9.0 is used for solving integer programming models in clustering and routing phases and Excel Visual Basic application is used for the local search and arranging the data for Gams on 2.20 GHz, 1GB RAM computer. Properties of benchmark problem set as number of linehaul customers, number of backhaul customers, capacity of vehicles, number of vehicles; the best known solution that are found in literature by this time, value of routing cost by proposed heuristic and the gap between the cost of proposed heuristic; CPU times of clustering phase, route phase, local search and total CPU time are presented in Table 1. Best known solutions are taken from [7]. Gap is calculated by following formulation.

$$gap = \frac{(value\ of\ proposed\ heuristic - best\ known\ solution)}{best\ known\ solution} * 100 \tag{15}$$

Average percentage gap of the solutions with the proposed heuristic is % 7.28. The solution of D1 is the best with % 0.05 gap and the solution of L3 is the worst with % 15.82 gap. The gaps of 4 benchmark problems are lower than %

1, the gaps of 13 problems are between % 1 and % 5, the gaps of 31 problems are between % 5 and % 10 and the gaps of 14 problems are between % 10 and % 16. In a sense, %77 of the benchmark problems gaps is lower than 0.1 and these are acceptable.

Table 1: Results of the benchmark problems

No	Name	N	M	C	K	Best known	Proposed heuristic	% Gap	CPU 1	CPU 2	CPU 3	Total CPU
1	A1	20	5	1550	8	229886	235370.64	2.39	0.156	0.590	0.016	0.762
2	A2	20	5	2550	5	180119	182775.2	1.47	0.031	0.295	0.031	0.357
3	A3	20	5	4050	4	163405	168715.4	3.25	0.093	0.421	0.016	0.53
4	A4	20	5	4050	3	155796	172311.5	10.6	0.234	0.124	0.047	0.405
5	B1	20	10	1600	7	239080	253834.6	6.17	1.89	0.826	0.031	2.747
6	B2	20	10	2600	5	198048	205067.2	3.54	0.62	0.669	0.031	1.32
7	B3	20	10	4000	3	169372	184738.6	9.07	0.656	0.186	0.047	0.889
8	C1	20	20	1800	7	249448	259403.9	3.99	1.765	0.528	0.031	2.324
9	C2	20	20	2600	5	215020	231539.28	7.68	1.734	0.341	0.031	2.106
10	C3	20	20	4150	5	199346	216451.41	8.58	0.14	0.514	0.078	0.732
11	C4	20	20	4150	4	195366	203982.76	4.41	0.796	0.374	0.078	1.248
12	D1	30	8	1700	12	322530	322704.53	0.05	0.921	1.229	0.016	2.166
13	D2	30	8	1700	11	316709	318475.7	0.56	3.296	0.869	0.031	4.196
14	D3	30	8	2750	7	239479	239931.81	0.19	0.515	0.59	0.031	1.136
15	D4	30	8	4075	5	205832	213785.8	3.87	0.718	0.373	0.047	1.138
16	E1	30	15	2650	7	238880	244987.74	2.56	0.281	0.59	0.047	0.918
17	E2	30	15	4300	4	212263	228689.73	7.74	1.875	0.2	0.078	2.153
18	E3	30	15	5225	4	206659	219091.85	6.02	0.39	0.373	0.14	0.903
19	F1	30	30	3000	6	263173	271140.70	3.03	177.062	0.512	0.16	177.734
20	F2	30	30	3000	7	265213	281690.50	6.21	59.543	0.951	0.22	60.714
21	F3	30	30	4400	5	241120	244810.90	1.53	0.187	0.56	0.28	1.027
22	F4	30	30	5500	4	233861	248678.24	6.34	0.218	0.778	0.41	1.406
23	G1	45	12	2700	10	306305	342391.25	11.78	162.843	0.84	0.047	163.73
24	G2	45	12	4300	6	245441	261374.07	6.49	6.046	0.686	0.063	6.795
25	G3	45	12	5300	5	229507	254100.94	10.72	2.109	0.608	0.109	2.826
26	G4	45	12	5300	6	232521	251717.55	8.26	0.14	0.686	0.203	1.029
27	G5	45	12	6400	5	221730	239617.26	8.07	2.812	0.56	0.281	3.653
28	G6	45	12	8000	4	213457	233044.42	0.65	0.171	1.176	0.156	1.503
29	H1	45	23	4000	6	268933	282430.11	5.02	3.921	6.39	0.219	10.53
30	H2	45	23	5100	5	253365	268541.94	5.99	0.593	0.679	0.484	1.756
31	H3	45	23	6100	4	247449	258918.24	4.63	0.312	0.5	0.562	1.374
32	H4	45	23	6100	5	250221	263745.07	5.40	0.281	1.717	0.64	2.638
33	H5	45	23	7100	4	246121	251675.26	2.26	0.203	1.03	0.734	1.967
34	H6	45	23	7100	5	249135	265165.90	6.43	0.187	1.138	0.64	1.965
35	I1	45	45	3000	10	350246	386984.79	10.49	29.343	1.654	0.094	31.091
36	I2	45	45	4000	7	309943	353217.06	13.96	53.59	1.29	0.265	55.145
37	I3	45	45	5700	5	294507	311194.65	5.67	33.14	2.458	0.5	36.098
38	I4	45	45	5700	6	295988	318030.18	7.45	2.671	0.496	0.453	3.62
39	I5	45	45	5700	7	301226	341089.78	13.23	0.968	0.887	0.422	2.277

No	Name	N	M	C	K	Best known	Proposed heuristic	% Gap	CPU 1	CPU 2	CPU 3	Total CPU
40	J1	75	19	4400	10	335006	363352.39	8.46	358.718	0.65	0.297	359.665
41	J2	75	19	5600	8	310417	332719.52	7.18	13.946	0.841	0.844	15.631
42	J3	75	19	8200	6	279219	317378.63	13.67	0.75	1.121	2.438	4.309
43	J4	75	19	6600	7	296553	323561.84	9.11	32.843	0.95	0.844	34.637
44	K1	75	38	4100	10	394376	420606.74	6.65	71.39	0.62	0.39	72.4
45	K2	75	38	5200	8	362130	382668.43	5.67	148.421	0.637	0.578	149.636
46	K3	75	38	5200	9	364693	389883.51	6.61	2.531	0.996	0.797	4.324
47	K4	75	38	6200	7	348950	361536.27	3.61	58.312	11.64	0.703	70.655
48	L1	75	75	4400	10	425772	461818.07	8.47	1000	2.465	1.031	1003.496
49	L2	75	75	5000	8	401228	463435.81	15.5	1000	14.58	1.203	1015.783
50	L3	75	75	5000	9	402720	466415.90	15.82	258.453	3.738	1.469	263.66
51	L4	75	75	6000	7	384637	415777.29	8.1	143.5	7.981	1.641	153.122
52	L5	75	75	6000	8	387928	432991.19	11.62	39.150	13.02	1.953	54.123
53	M1	100	25	5200	11	398593	439396.75	10.24	101.578	0.542	0.328	102.448
54	M2	100	25	5200	10	396917	434896.47	9.57	1000	0.95	0.297	1001.247
55	M3	100	25	6200	9	376309	411726.12	9.41	254.531	2.218	0.375	257.124
56	M4	100	25	8000	7	348418	373525.23	7.21	17.859	1.091	0.469	19.419
57	N1	100	50	5700	11	408101	466378.20	14.28	16.156	1.495	2.375	20.026
58	N2	100	50	5700	10	408066	440112.39	7.85	1000	2.763	1.75	1004.513
59	N3	100	50	6600	9	394338	426869.67	8.25	614.156	4.355	3.453	621.964
60	N4	100	50	6600	10	396055	435184.91	9.88	3.562	5.479	3.781	12.822
61	N5	100	50	8500	7	373477	431326.66	15.49	84.43	99.807	5.39	189.627
62	N6	100	50	8500	8	374691	423357.66	12.99	10.875	84.141	5.5	100.516

Average CPU time of clustering phase is 109.41 seconds, average CPU time of routing phase is 4.77 seconds, average CPU time of local search is 0.74 seconds and average total CPU time of the proposed heuristic is 114.94 seconds. Minimum CPU time is belong to A2 with 0.031 seconds for clustering phase, A4 with 0.124 seconds for routing phase, A1 with 0.016 for local search and A2 with 0.357 for total. L1, L2, M1 and N2 with 1000 seconds for clustering phase, N5 with 99.807 for routing phase, N6 with 5.5 seconds for local search and L2 with 1015.783 for total have the maximum CPU time. This is to say that the heuristic takes time more than 17 minutes for large scale problems and it find solution within a few seconds for small and medium scale problems. Clustering phase needs more time than other phases in the heuristic.

4. Conclusion

The proposed heuristic with two phases is similar to cluster-first route second heuristics. Customers are assigned to vehicles by proposed integer programming in clustering phase. The proposed integer programming model is firstly generated and used for clustering phase while other studies are used some algorithms for this purpose. Also there is no need to define seed customers at the beginning of the solution procedure because the integer programming model is defined them among all customers while seed customers should be defined by decision maker in other algorithms. Each cluster is routed as TSPB in routing phase. The integer programming model that is used routing phase is modified from Goetschalckx and Jacobs-Blecha's [19] model for VRPB by removing vehicle capacity and assigning customers to vehicles constraints.

The results that are generated by proposed heuristic on the test problems are presented. According to best known solutions, the results are acceptable. Because the closeness between best known solution and proposed heuristic is over 0.9 for %77 of benchmark problems. CPU time is a few seconds for small and medium scale problems and is around 17 minutes for large scale problems. Most of the CPU time is for clustering phase. So CPU time for finding solution by the proposed heuristic is admissible.

The main contributions of the proposed heuristic are to propose an integer programming model for the clustering phase, to form an integer programming model for TSPB by modifying VRPB model, to use only integer programming for both clustering and routing phase and to combine integer programming models and local search.

References

1. Ai, T. J. and Kachitvichyanukul, V., 2009, “Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem” Computers & Industrial Engineering, 56, 380-387.
2. Alfa, A. S., Heragu, S. S. and Chen, M., 1991, “A 3-opt based simulated annealing algorithm for vehicle routing problems” Computer & Industrial Engineering, 21, 635-639.
3. Baker, B. M. and Ayechew, M., A., 2003, “A genetic algorithm for the vehicle routing problem” Computers & Operations Research, 30, 787-800.
4. Baker, B. M. and Sheasby, J., 1999, “Extensions to the generalized assignment heuristic for vehicle routing” European Journal of Operational Research, 119, 147-157.
5. Barbarosoglu, G. and Ozgur, D., 1999, “A tabu search algorithm for the vehicle routing problem” Computers and Operations Research, 26, 255-270.
6. Bin, Y., Zong-Zhen, Y. and Baozhen, Y., 2009, “An improved ant colony optimization for vehicle routing problem” European Journal of Operational Research, 196, 171-176.
7. Brandão, J., 2006, “A new tabu search algorithm for the vehicle routing problem with backhauls” European Journal of Operational Research, 173, 540-555.
8. Bullnheimer, B., Hartl, R. F., and Strauss, C., 1999, “An improved ant system algorithm for the vehicle routing problem” Annals of Operations research, 89, 319-328.
9. Cardeau J-F. and Maischberger M., 2012, “A parallel iterated tabu search heuristic for vehicle routing problems”, Computers and Operations Research, 39, 2033-2050.
10. Clarke, G. and Wright, J. V., 1964, “Scheduling of vehicles from a central depot to a number of delivery points” Operations Research, 12, 568-581.
11. Dantzig, G. B. and Ramser, J. H., 1959, “The Truck dispatching problem” Management Science, 6(1), 80-91.
12. Deif, I.,and Bodin, L., 1984, “Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauls” The Babson Conference on Software Uses in Transportation and Logistics Management, Babson Park, 75-96.
13. Dijkstra, E. W., 1959, “A note on two problems in connection with graphs” Numerische Mathematik, 1, 269-271.
14. Fisher, M. L. and Jaikumar, R., 1981, “A generalized assignment heuristic for the vehicle routing problem” Networks, 11, 109-124.
15. Gajpal, Y. and Abad, P. L., 2009, “Multi-ant colony system (MACS) for a vehicle routing problem with backhauls” European Journal of Operational Research, 196, 102-117.
16. Gendreau, M., Hertz, A. and Laporte, G., 1994, “A tabu search heuristic for the vehicle routing problem” Management Science, 40, 1276-1290.
17. Gillett, B. E. and Miller, L. R., 1974, “A heuristic algorithm for the vehicle dispatch problem” Operations Research, 22, 340-349.
18. Goetha, S., Poonthalir, G. and Vanathi, P. T., 2010, “A hybrid particle swarm optimization with genetic operators for vehicle routing problem”, Journal of Advances in Information Technology, 1(4), 181-188.
19. Goetschalckx, M. and Jacobs-Blecha, C., 1989, “The vehicle routing problem with backhauls” European Journal of Operational Research, 42(1), 39-51.
20. Goetschalckx, M. and Jacobs-Blecha, C., 1993, “The vehicle routing problems with backhauls: Properties and solution algorithms”: Technical Report- MHRC-TR.88-13, Georgia Instıtute of Technology.
21. Laporte, G. and Semet, F., 2001, “Chapter 5: Classical Heuristics for the Capacitated VRP” appears in The Vehicle Routing Problem, Toth, P. and Vigo, D. (eds.), SIAM, Bologna, 109-128.
22. Lin, S., 1965, “Computer solutions of the traveling salesman problem” Bell System Technical Journal, 21, 498-516.
23. Lee, C., Lee, Z., Lin, S. and Ying, K., 2010, “An enhanced ant colony optimization (EACO) applied to capacited vehicle routing problem”, Applied Intelligence, 32(1), 88-95.
24. Lin, S., Lee, Z., Ying, K. and Lee, C., 2009, “Applying hybrid meta-heuristics for capacitated vehicle routing problem” Expert Systems with Applications, 36, 1505-1512.

25. Liu C., Li L. and Wu J., 2011, “An effective tabu search for the vehicle routing problem with backhaul”, Smart Materials and Intelligent System, 143, 1415-1419.
26. Marikinas, Y., Marinaki, M. and Dounias, G., 2010, “A hybrid particle swarm optimization algorithm for the vehicle routing problem” Engineering Applications of Artificial Intelligent, 23, 463-472.
27. Marinakis, Y., and Marinaki, M., 2010, “A hybrid genetic-particle swarm optimization algorithm for the vehicle routing problem” Expert System with Applications, 37, 1446-1455.
28. Mazzeo, S. and Loiseau, I., 2004, “An ant colony algorithm for the capacitated vehicle routing” Electronic Notes in Discrete Mathematics, 18, 181-186.
29. Mingozzi, A. and Giorgi, S., 1999, “An exact algorithm for the vehicle routing problem with backhauls” Transportation Science, 33(3), 315-329.
30. Nazif H. and Lee L. S., 2012, “Optimized crossover genetic algorithm for capacitated vehicle routing problem”, Applied Mathematical Programming, 36, 2110-2117.
31. Osman, I. H., 1993, “Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems” Annals of Operations Research, 41, 421-451.
32. Reimann, M., Doerner, K. and Hartl, R.F., 2004, “D-Ants: Saving based ants divide and conquer the vehicle routing problem” Computers & Operations Research, 31, 563-591.
33. Ropke, S. and Pisinger, D., 2006, “A unified heuristic for a large class of vehicle routing problems with backhauls” European Journal of Operational Research, 171, 750-775.
34. Salhi, S. and Nagy, G., 1999, “A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling” Journal of Operational Research Society, 50, 1034-1042.
35. Tavakkoli-Moghaddam, R., Saremi, A. R. and Ziaee, M. S., 2006, “A memetic algorithm for a vehicle routing problem with backhauls” Applied Mathematics and Computations, 181, 104-1060.
36. Toth, P. and Vigo, D., 2003, “The granular tabu search and its application to the vehicle routing problem” INFORMS Journal of Computing, 15, 333-346.
37. Toth, P. and Vigo, D., 1999, “A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls” European Journal of Operational Research, 113, 528-543.
38. Toth, P. and Vigo, D., 1997, “An exact algorithm for the vehicle routing problem with backhauls” Transportation Science, 31(4), 372-385.
39. Toth, P. and Vigo, D., 1996, “A heuristic algorithm for the vehicle routing problem with backhauls” appears in Advanced Methods in Transportation Analysis, Bianco, L. and Toth, P. (eds.), Springer Verlag, Berlin, 585-608.
40. Tütüncü, G. Y., Carreto, C. A. C. and Baker, B. M., 2009, “A visual interactive approach to classical and mixed vehicle routing problems with backhauls” The International Journal of Management Science, 37, 138-154.
41. Van, B., 1995, “A. Improvement heuristics for the vehicle routing problem based on simulated annealing” European Journal of Operational Research, 86, 480-490.
42. Wang, Z. and Wang, Z., 2009, “A novel two-phase heuristic method for vehicle routing problem with backhauls” Computer and Mathematics with Applications, 57, 1923-1928.
43. Xu, J. and Kelly, J. P., 1996, “A network flow-based tabu search heuristic for the vehicle routing problem” Transportation Science, 30, 379-393.
44. Zacharia, E. E. and Kiranoudis, C.T., 2012, “An effective local search approach for the vehicle routing problem with backhauls”, Expert Systems with Applications, 39, 3174-3184.
45. Zhang, X. and Tang, L., 2009, “A new hybrid ant colony optimization algorithm for the vehicle routing problem” Pattern Recognition Letters, 30, 848-855.