Stochastic multi-depot capacitated vehicle routing problem with pickup and delivery: heuristic approaches

Brenner H. O. Rios[1], **Eduardo C. Xavier**[1], **Flávio K. Miyazawa** [1], **Pedro Amorim** [2]

[1]Institute of Computing – University of Campinas (UNICAMP)
Campinas – SP – Brazil

[2]Faculty of Engineering – University of Porto
Porto, Portugal.

brenner@students.ic.unicamp.br, {ecx,fkm}@ic.unicamp.br,

amorim.pedro@fe.up.pt

Abstract. *We present a natural probabilistic variation of the multi-depot vehicle routing problem with pickup and delivery. We denote this variation by Stochastic multi-depot capacitated vehicle routing problem with pickup and delivery (SMCVRPPD). We present an algorithm to compute the expected length of an a priori route under general probabilistic assumptions. To solve the SMCVRPPD we propose an Iterated Local Search (ILS) and a Variable Neighborhood Search (VNS). We evaluate the performance of these heuristics on a data set adapted from TSPLIB instances. The results show that the ILS is effective to solve SMCVRPPD.*

1. Introduction

After several decades of development, the vehicle routing problem (VRP) has become an important content in logistics management research, and has been widely used in transportation system, logistics distribution system and express delivery system [Cao and Yang 2017]. Several versions of the vehicle routing problem have been proposed, such as the capacitated vehicle routing problem (CVRP), multi-depot vehicle routing problem (MDVRP) and vehicle routing problem with pickups and deliveries (VRPPD) [Oyola et al. 2017]. To better describe the real world, stochastic data may be included in the model. In this case a new problem arises, stochastic multi-depot capacitated vehicle routing problem with pickup and delivery (SMCVRPPD). We provide a description of the SMCVRPPD in Section 2. Section 3 presents the an algorithm to compute the expected length of an *a priori* route. Section 4 describes the proposed heuristic and computational experiments. Section 5 presents the conclusions of this work.

2. Problem description

In this work, SMCVRPPD is defined as follows. Let $G = (V, E)$ be a complete undirected graph, where $V = \{v_1, \ldots, v_n\}$ is the vertex set and $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ is the edge set. With each edge (v_i, v_j), it is associated a non-negative cost or distance d_{ij}. A subset of vertices $D = \{v_1, \ldots, v_m\}$ represents the depots, and the remaining vertices $V' = \{v_{m+1}, \ldots, v_n\}$ corresponds to pickup and delivery points. Let $w = |V'|/2$, then w vertices are pickup points and w vertices are delivery points. Each pickup point v_i is associated with a unique delivery point v_{i+w}, and vice versa, for $m + 1 \leq i \leq m + w$.

There are m identical vehicles of unlimited capacity such that each one is located in a single depot. Each vehicle leaves its depot, serves a subset of pickup and delivery vertices and returns to its depot, forming a cycle (or route). The problem consists in determining a set of m vehicle cycles of minimal total cost considering the following constraints: a) each cycle starts and ends at the corresponding vehicles depot; b) each $v \in V'$ is visited exactly once by one vehicle c) each pair of pickup and delivery points, e.g $\{v_i, v_{i+w}\}$ for $m+1 \leq i \leq m+w$, must belong to the same cycle and d) each cycle has an orientation where each pickup vertex in this cycle appears before its delivery pair.

The SMCVRPPD has one type of uncertainty: stochastic pickup and delivery points. Each subset $\{v_i, v_{i+k}\} \in V'$, for $m+1 \leq i \leq m+k$, has a probability p_i of being present. When pickup point v_i is absent, delivery point v_{i+k} is also absent, and its demand ξ_i is equal to zero otherwise is equal to 1. In this work, we assume all demands are unitary (1 or 0), discrete and independent. In a *first stage*, a set of cycles satisfying conditions a)- d) of the SMCVRPPD are determined. The presence or absence of $\{v_i, v_{i+k}\}$ is revealed at the latest upon leaving the preceding vertex of v_i, but the positive demand of every remaining customer becomes known only when the vehicle arrives at the delivery's location. In a *second stage*, the first stage routes are followed as planned, with the following exceptions: 1) any absent customer is skipped and 2) whenever the remaining vehicle capacity is not enough to carry current node demand, the vehicle follows a strategy, in either case, route failure is said to occur and a strategy used to deal with the problem failure. We use the following strategy: once the vehicle reaches its maximum capacity, a penalty is paid for the transportation of excess cargo by an outsourced service, and the vehicle continues the original route. The SMCVRPPD consists of designing a first stage solution to minimize the expected cost of the second stage solution.

3. The expected cost of an *a priori* route

Given a priori computed route $s = (v_0, v_1, \ldots, v_{2q}, v_0)$, where v_0 is a depot, let l_s be the cost/length of s. Our goal is to compute the expected length $E[l_s]$ of route s, given that during its execution, each pair $\{v_i, v_{i+w}\}$ of pickup and delivery points in this route have a probability of occurring during s's execution. We may also refer to node v_i as r_i, and v_{i+w} as c_i. Let $P(v_i)$ be the probability that node v_i appears in s, therefore $P(r_i) = P(c_i)$. In this work we use the following strategy, each time the maximum capacity Q of the vehicle is reached, a penalty must be paid. Such penalty represents the cost of travel from the pickup point where the vehicle was full to its associated delivery point. Then the vehicle load is updated to $Q - 1$ and the vehicle continues the original route. To compute the

Algorithm 1: Expectation

```
$exp = 0$ ;
for $s_i \subseteq S$ do
    $prob := 1$, $cost :=$ Cost of the route $s_i$ using strategy with penalty
    for $(r_j, c_j) \in s_i$ do
        $prob = prob \times P(r_j)$
    for $(r_j, c_j) \notin s_i$ do
        $prob = prob \times (1 - P(r_j))$
    $exp = exp + cost \times prob$
```

expected cost $E[l_s]$ of a route s we use Algorithm 1. Let s_i be a valid route generated from the route s, e.g., s_1 = (v_0) and $s_2 = (v_0, r_1, c_1, v_0)$ are valid routes. In s_1 no pair of pickup and delivery appear in the route. In s_2 only the pair (r_1, c_1) appears. S denotes the set of valid routes generated from s. There are 2^q valid routes generated from s.

4. Heuristics approaches

We present the local search operators, perturbation operators, and heuristics used by VNS and ILS. We adapt the algorithms VNS [Rios et al. 2020b] and ILS [Rios et al. 2020a] for the SMCVRPPD. This means that the operators are executed following the sequence of the original algorithms.

4.1. Initial solution generation

The method employed for building a feasible initial solution is based in the work of [Kuo and Wang 2012], so it is generated by following two steps. The first step is called *nodes assignment*. Each pair of pickup and delivery is assignment to one of the depots. After all vertices have been assigned to depots, the second step, called *nodes sequencing*, decides the service sequence of the pickup and delivery nodes.

4.2. Local search

A set $\{N^1, \ldots, N^6\}$ of six neighborhood operators were used by the proposed heuristics. All operators are exhaustively executed. These operators are adapted in such a way that they preserve feasibility. The list of neighborhoods considered are:

- **Shift(1,0)** – N^1 – A pickup and delivery pair r, c is removed from a route t_1 and each one is moved to the best position in route t_2 keeping the feasibility of the solution.
- **Swap(1,1)** – N^2 – An exchange between a pair r_1, v_1 from a route t_1 and another pair r_2, v_2 from route t_2. Each vertex of the pairs are inserted in the best possible position while maintaining the feasibility of the solution.
- **Mix-Shift(1,0)** – N^3 – This operator is similar to the Shift(1,0) operator with the difference that now it is allowed a movement within its own route.
- **2-opt** – N^4 – Two nonadjacent arcs are removed and other two are added to form a new route. We only consider movements that do not break the constraints of the problem.
- **3-opt** – N^5 – Three nonadjacent arcs are removed and other two are added to form a new route. We only consider movements that do not break the constraints of the problem.
- **Reverse** – N^6 – This operator reverses the direction of the route. Then swaps are performed between each pair of pickup and delivery.

4.3. Perturbation Mechanism

A set P of two perturbation mechanisms were adopted in the heuristics. Every time the *perturb()* function is called one of the following operators is randomly selected and applied.

- **Double-Swap** – P^1 – Two Swap(1,1) operators are performed in sequence.
- **Depot Exchange** – P^2 – The depot exchange operator select two depots at random, and exchange their routes.

Instance				ILS			VNS		
Name	$\lvert V\rvert$	D	Q	Best Cost	Avg. Cost	Time (s)	Best Cost	Avg. Cost	Time (s)
bayg29a	29	3	4	**378.60**	652.96	23.51	**378.60**	670.93	25.10
bayg29b	29	9	4	**3185.55**	4388.36	10.77	4055.68	4661.49	12.38
bayg29c	29	9	2	**1717.82**	2134.55	3.84	1836.92	2161.74	4.14
dantzig42a	42	8	4	**272.98**	348.09	3.26	288.99	350.88	3.55
dantzig42b	42	4	4	**91.24**	141.94	1539.08	107.75	144.41	1558.08
dantzig42c	42	10	2	**304.77**	413.34	2.55	354.52	419.87	2.85
eil51a	51	3	4	**60.41**	117.29	8282.83	106.39	131.08	8382.83
eil51b	51	7	5	**79.67**	112.04	123.72	86.43	112.71	134.72
eil51c	51	9	3	**136.15**	193.84	8.34	159.49	196.17	8.45

5. Computational experiments

We conducted experiments using a data set derived from three TSPLIB instances (*bayg29, dantzig42* and *eil51*). For each of these instances, n vertices in the interval of [2, 10] were randomly selected to be depots. The capacity of each vehicle Q was randomly generated in the interval [2,10]. A random matching was performed among the other vertices to create pickup and delivery pairs. The probability of presence of each pickup and delivery pair was chosen uniformly in the interval $[0, 1]$. We generate 9 test instances. The proposed heuristic (VNS and ILS) were coded in C++ and all experiments were run on a Linux operating system with 3 GB memory and Intel Core i5 2.54x4 Ghz processor. Computational times reported here are in CPU seconds on this machine. Ten independent runs of the algorithms were performed for each test case.

The results show that the ILS was superior for all instances tested. Our approach can be used as benchmark for future research in this area. The SMCVRPPD can be further generalized to handle more practical constraints, e.g., time windows and stochastic demands.

References

Cao, W. and Yang, W. (2017). A survey of vehicle routing problem. In *MATEC Web of Conferences*, volume 100, page 01006. EDP Sciences.

Kuo, Y. and Wang, C.-C. (2012). A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. *Expert Systems with Applications*, 39(8):6949–6954.

Oyola, J., Arntzen, H., and Woodruff, D. L. (2017). The stochastic vehicle routing problem, a literature review, part ii: solution methods. *EURO Journal on Transportation and Logistics*, 6(4):349–388.

Rios, B., Xavier, E. C., Miyazawa, F. K., and Amorim, P. (2020a). Multi-depot multiple tsp with pickup and delivery: an vns approach. In *ANAIS DO LII SIMPóSIO BRASILEIRO DE PESQUISA OPERACIONAL (SBPO).*

Rios, B., Xavier, E. C., Miyazawa, F. K., and Amorim, P. (2020b). Stochastic multi-depot vehicle routing problem with pickup and delivery: an ils approach. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 307–315. IEEE.