DOI: https://doi.org/10.29121/granthaalayah.v7.i3.2019.976

INTERNATIONAL JOURNAL OF RESEARCH – GRANTHAALAYAH
A knowledge Repository

Science

CAPACITATED VEHICLE ROUTING PROBLEM

Ibrahim A.A[*1], Lo N. [2], Abdulaziz R.O [3], Ishaya J.A [4]
[*1, 2, 4] Department of Mathematical Science, African Institute for Mathematical Science, Senegal
[3] Department of Energy Engineering, PAUWES, University of Tlemcen, Algeria

Abstract

Cost of transportation of goods and services is an interesting topic in today’s society. The Capacitated vehicle routing problem, which is been consider in this research, is one of the variants of the vehicle routing problem. In this research we develop a reinforcement learning technique to find optimal paths from a depot to the set of customers while also considering the capacity of the vehicles, in order to reduce the cost of transportation of goods and services. Our basic assumptions are; each vehicle originates from a depot, service the customers and return to the depot, the vehicles are homogeneous. We solve the CVRP with an exact method; column generation, goole’s operation research tool and reinforcement learning and compare their solutions. Our objective is to solve a large-size of vehicle routing problem to optimality.

Keywords: Vehicle Routing Problem; CVRP; Column Generation; Google’s OR tool, Reinforcement Learning; Python; Gourbi.

Cite This Article: Ibrahim A.A, Lo N., Abdulaziz R.O, and Ishaya J.A (2019). “CAPACITATED VEHICLE ROUTING PROBLEM.” *International Journal of Research - Granthaalayah*, 7(3), 310-327. https://doi.org/10.29121/granthaalayah.v7.i3.2019.976.

1. Introduction

One of the major challenges people face in their daily activities is the problem of transportation. The transportation of goods and services is an important topic of concern in today’s society. A large sum of money is spent daily on fuel, goods and service delivery, equipment maintenance and so on. This is where the knowledge and technique of Operations Research (OR) comes to play. If the available resources is known, one can employ the techniques of OR. According to [1], the use of computerized techniques in solving transportation problem most times often leads to about 5% − 20% savings on transportation cost. Therefore, planning of distribution process, research and studying OR-techniques is worthwhile and will save some transportation cost.

In transportation, one major area that has received lots of attention over the years which relates to OR is the Vehicle Routing Problem (VRP). A lot of research work has been carried out and much progress and improvement has been made ever since the first article on “truck-dispatching-problem” was published by [2]. Since the introduction of OR in solving transportation problems, success has been recorded especially with the use of optimization techniques. Several techniques on exact methods and the heuristics with strong formulations has been proposed and developed in solving the VRP.

The transportation sector was responsible for 28% of carbon IV oxide (CO_2) generated in the European Union in 1998 and 84% of the CO_2 was emitted from road transportation alone [3]. Also [3] predicted that by the year 2010, the CO_2 emitted from transportation was expected to increase by 50%. Hence, an improvement on the planning techniques could help reduce the CO_2 emitted and reduce the cost of transportation. Consequently, the knowledge and application of optimization technique in handling transportation problem is not only studied in higher institutions but has been sought after in the industry to solve real-life problems.

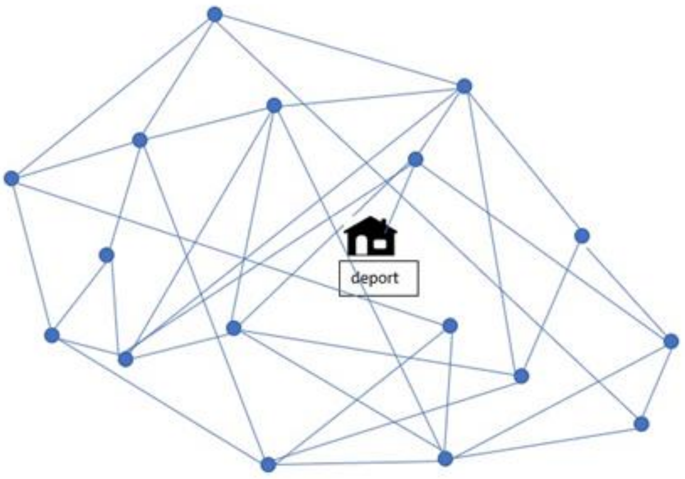


Illustration 1: Initial graph showing locations and paths between them

We shall model real-life problem using a graph; which consist of vertices and edges. Where the vertices represent the known locations or cities and the edges denotes the path between two locations.

The concept of VRP was first proposed by [2] and a mathematical programming formulation and algorithm method for VRP were also developed in this study. This problem generalizes the famous and common Traveling Salesman Problem (TSP) which is one of the simplest routing problems. According to [4], the TSP involves finding the optimal/shortest route that connects all routes exactly once and return to the starting node from a given a set of finite route and also measuring the distance between them.

The image of the right shows a graph with 18 cities and 4 vehicles, all starting and returning to a single depot. Since all the vehicles are homogeneous, a route (defined path) has been designed for each of these vehicles.

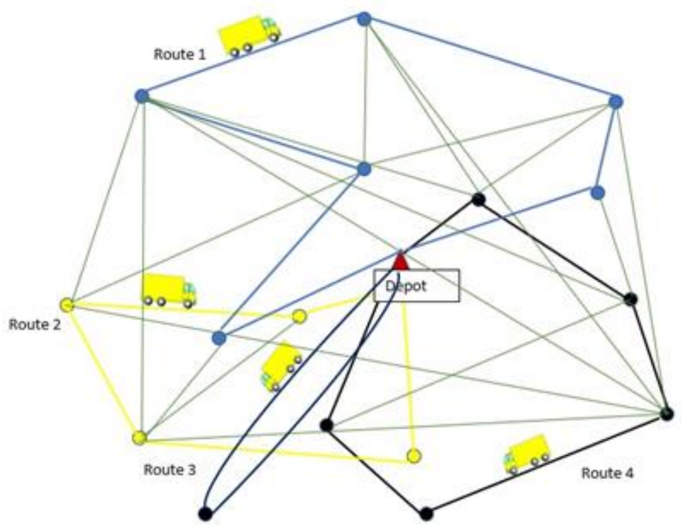


Illustration 2: Final graph showing optimal routes

Due to the set of several available routes, VRP is a computationally difficult problem even though many algorithms (such as heuristic and exact algorithm) have been proposed over the years. The challenging task is how to produce a solution that will be fast and reliable. The objective of VRP is to find an optimal route from a set of routes in which each vehicle starting from a depot (say Node A) and connect all customers, and then must return to the depot A [5].

The VRP is a combinatorial optimization and integer programming problem which finds optimal path in order to deliver goods and services to a finite set of customers. The VRP has been a problem for several decades and one of the most studied problem in logistics engineering, applied mathematics and computer science, and which is described as finding optimal routes for a fleet of vehicles to serve some scattered customers from depot [6].

In other words, VRP involves a vehicles delivering goods (and services) to multiple customer nodes and when the vehicle runs out of items it must return to the deport to pick additional goods in other to continue with supplying the customers. A vehicle will visit a customer exactly once and the vehicle must return to the depot where it started from.

In this research work, our goal is to find optimal set of routes for some vehicles delivering goods or services to some known locations. To achieve this goal, we shall formulate a mathematical model for the CVRP problem. Then, solve this model using three techniques and compare the solutions of these techniques using small-size and large-size problems in other to determine a technique capable of solving large-size problem.

The VRP consists of several variants which but not limited to; TSP, Multiple TSP, Capacitated VRP, VRP with time windows, dynamic VRP, pickup and delivery VRP, periodic VRP and so on. These variants have several application which include; DVRP is applicable in the courier services [7]. Mlk-collection problem [8], and so on.

Several mathematicians and engineers had done research on VRP and its variants. Some researchers came up with significant ideas which has really contributed to solving VRP and transportation problem in general.

The first article was introduced by [2], some trial problems were calculated but no practical application was made. Based on the first article, [9] developed a saving-based heuristics algorithm that enables rapid selection of optimal or near-optimal route. More work on this technique can be found in [9].

Column Generation (CG), an exact approach for solving VRP was used and reported successful in solving VRPTW [10, 11]. [12] solved a CVRP problem using an exact method based on set-partitioning formulation of the problem. The proposed method was effective but has limited solving power of customers to about 100.

[13] presented a reinforcement learning framework used to solve the VRP. The model was applied on CVRP and the approach performed very well on medium-sized problem in terms of the solution quality with computational time (after training). Advantage of this method is that it scales well when the problem-size is increasing.

1.1. Aims and Objectives

In this research work, our goal is to find optimal set of routes for some vehicles delivering goods or services to some known locations. Further, the capacity of these homogeneous vehicles must not be exceeded.

To achieve this goal, we shall;

- Formulate a mathematical model for the CVRP problem.
- Solve this mathematical model using three techniques
- Compare the solutions of these techniques using small-size and large-size problems in order to determine a technique capable of solving large-size problem.
- Apply our technique on large-size problem.

2. Materials and Methods

We shall define some parameters here. All vehicles will originate and end at the depot, while each of the customer is visited exactly once. Let us define the following:

- $C = \{v_1 , v_2 , . . . , v_m \}$: represent the set of m-customers to be considered.
- L: denote the fleet of available vehicles in a single depot. All vehicles considered are homogeneous, and we have n-vehicles.
- Q: is the maximum capacity of a vehicle, which limits the number of customers to be visited before returning to the depot.

The vehicle routing problem is a directed graph *G(V, E)* with a cost-matrix, *C* where

- $V = \{v_0 , v_1 , . . . , v_m , v_{m+1}\}$ is the set of vertices associated with C. The vertices $\{v_0 , v_{m+1}\}$ represent the depot, i.e $v_0 = v_{m+1}$ and $\{v_1 , . . . , v_m \}$ represent m-customers.

- E = {(v_i , v_j) | $0 \leq i, j \leq$ m, i = j} is a set of |V| ∗ (|V| − 1) directed routes/edges between the vertices. If in both directions the distance between two vertices are identical, we then add the (i < j) restriction, and this is the symmetric variant.
- C = (c_{ij}) is a cost-matrix and $c_{ij} \geq 0$ is the corresponding distance of edges (v_i , v_j), the diagonal of the matrix i.e c_{ii} = 0 always. Depending on whether the VRP variant in consideration is symmetric or not, c_{ij} = c_{ji} . The triangle inequality is assumed to hold generally, i.e $c_{ij} \leq c_{ik} + c_{kj}$ and ($0 \leq$ i, j, k $\leq$ m).

Furthermore, we need to define some important terms in this VRP problem;

- Ri = (v0i, v1i, v2i, v3i , . . . , viki , viki+1) is a vector of the route of vehicle i which start and end at the depot, with v0i = viki+1 = v0 , $v_j^i \neq v_l^i$, 0 ≤ j < l ≤ ki , and ki is the length of route Ri .
- S = {R_1, . . . , R_n } is the set of route which represent the VRP solution instance.
- C (R_i) = $\sum_{j=0}^{k_i} C\left(v_j^i, v_{j+1}^i\right)$is the cost of route Ri.
- • C(S) = $\sum_{i=1}^{n} C\left(R_i\right)$is the total cost of solution S which satisfies;
- $R_i \cap R_j$ = {v_0 } ∀R_i , R_j , (1 ≤ i, j ≤ n, $i \neq j$) and .$\cup_{i=1}^{n} R_i = V$

in order for each customer to be served once. The route vectors is treated here as a set. The goal of the VRP is to minimize the *C(S)* on the graph *G(V, E).*

G is the graph which contains |*E*|+*2* vertices, and the customers ranges from (1, 2, . . . , m). The starting and returning depots are denoted by 0 and *m + 1* respectively. Earlier in this section, we introduced the vehicle routing problem which we have now defined. However, the problem is not all about visiting the customers, there is more to their demands. In the following definitions, we shall specify these additional demands of the customers:

- demand; d = (d_0 , . . . , d_m , d_{m+1}) with $d_i > 0$ and *m* is the total number of customers which is a vector of the demands of customer, the demand of the depot is denoted by d_0 ; d_0 = d_{m+1} = 0 always.
- service time; *δ* is a function of service time: time to unload all the goods at customer v_i , i = 1, 2, . . . , m. Often times, *δ* is dependent on the size of the customer's demand. Henceforth, we shall use these notations as the same henceforth, $\delta_i = \delta(v_i)$.
- Let us define our decision variable as y_{ij} = 1 if *(i, j)* is a route and 0 otherwise.

The problem definition will be based on the following assumptions;

- The capacity constraints of all the vehicles are observed.
- Each customer can be served by only one vehicle.
- Each and every route starts at vertex 0 and ends at vertex (*m + 1*).

The mathematical formulation of Capacitated Vehicle Routing Problem is stated below.

We start with the objective function;

$$min \sum_{i=0}^{m+1} \sum_{j=0}^{m+1} C_{ij}\, y_{ij}, \qquad (2.0.1)$$

which minimize the total travel cost by vehicle. This function is subjected to several constraints.

Subjected to:

$$\sum_{\substack{j=1 \\ j\neq 1}}^{m+1} y_{ij} = 1, \forall i = 1,2,\ldots,m \qquad (2.0.2)$$

restrict each customer to be visited and served by only one vehicle

$$\sum_{\substack{j=1 \\ j\neq 1}}^{m+1} y_{ij} = 1, \qquad (2.0.3)$$

$$\sum_{\substack{i=0 \\ i\neq h}} y_{ih} - \sum_{\substack{j=1 \\ j\neq h}} y_{hj} = 0, \qquad (2.0.4)$$

$$\sum_i y_{i,m+1} = 1, \qquad (2.0.5)$$

constraints (2.0.3), (2.0.4) and (2.0.5) ensure that each and every vehicle must originate from starting depot 0, pass through various destinations of demands and return to end depot *m + 1*.

$$x_j \geq x_i + d_j y_{ij} - Q\big(1 - y_{ij}\big), \forall i, j = \{0,1,\ldots,m+1\} \qquad (2.0.6)$$

$$d_i \leq x_i \leq Q, \forall i = \{0,1,,\ldots,m+1\}, \qquad (2.0.7)$$

constraints (2.0.6) and (2.0.7) ensures the capacity constraint is observed.

$$y_{ij} \in \{0,1\}, \forall i, j = \{0,1,,\ldots,m+1\}, \qquad (2.0.8)$$

and constraint (2.0.8) indicate integrality constraints.

Note that subtours are avoided in the solution with constraint (2.0.6) that is, cycling paths which do not pass through the depot. Constraints (2.0.6) and (2.0.7) advantage in this problem is that in terms of our customers, the formulation has a polynomial number of constraints.

However, the Linear Programming (LP) relaxation of this formulation (model) generate a lower bound which is known to be weak when compared to other models. Many researchers and authors emphasized on capacity constraints that produce a better lower bounds, although the constraints increases exponentially in terms of number of customer thereby requiring the application of branch and cut (BAC) technique [14].

2.1. Column Generation (CG)

In order to solve our objective function (2.0.1), we shall apply CG technique. This technique is efficient in solving large linear program problems [15].

Restricted Master Problem: The Restricted Master Problem (RMP) is a set-partitioning problem. Some routes have the potential to improve the objective function, the RMP considers these paths which were added to the set of routes due to their potential to improve the objective function.

$$min \sum_{k \in \mathbb{R}} c_k \, y_k, \tag{2.1.1}$$

subjected to:

$$\sum_{k \in \mathbb{R}} a_{ik} \, y_k = 1, \forall i \in V - \{0\} \tag{2.1.2}$$

$$y_k \geq 0, \forall k \in \mathbb{R} \tag{2.1.3}$$

Pricing Problem: The second part of CG is the pricing-problem which is a sub-problem following the RMP to identify and generate new routes and column that will enter the set of routes (variables) in the RMP.

$$min \sum_{i \in V} \sum_{j \in V} \left(c_{ij} - \pi_i\right) yk, \tag{2.1.4}$$

subjected to:

$$min \sum_{i \in \mathbb{C}} d_i \sum_{j \in \mathbb{N}} y_k \leq Q, \tag{2.1.5}$$

$$y_k \in \{0,1\} \tag{2.1.6}$$

2.2. Google's OR Tool (OR-Tool)

The second technique to be used is the Operations Research Tools (OR-Tool). OR-Tool is an open source software suitable for solving optimization problems. The OR-Tools is suitable in solving routing problem, constraint programming, flows problem, integer programing and so on [16]. The advantage of this software to us is that it enables us to find optimal tour and its length for routing problem using python. It computational time is usually very fast compared to other techniques. Its solutions are usually near-optimal when compared with exact method.

The OR-Tool computes distance between two points; (x_1, y_1), (x_2 , y_2), using the Manhattan distance which sum up the absolute distance of x and y coordinates respectively. This can be obtained mathematically as;

$$C = |x_1 - x_2| + |y_1 - y_2| \tag{2.2.1}$$

We convert the OR-Tools formula for computing the cost of transportation by computing the distance between two coordinates; (x_1 , y_1), (x_2 , y_2), i.e using the Euclidean formula;

$$C = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{2.2.2}$$

2.3. Reinforcement Learning

Reinforcement learning (RL) is the learning of what-to-do, how to map situations to actions in order to maximize a numerical reward signal. The learner is not explicitly told the actions to take, but instead discover which actions yield more rewards by trying them. RL provides a mathematical framework suited to solving games. In RL, Markov Decision Process (MDP), tool for modeling artificial intelligence agents that interacts with environment that offers rewards on completion of some certain actions, is the central mathematical concept.

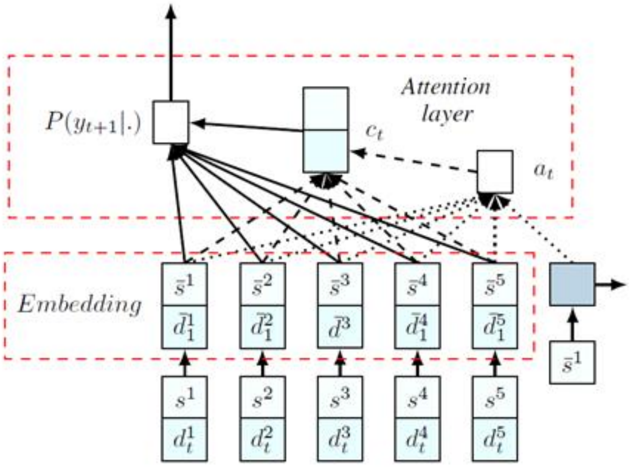

Attention layer
$P(y_{t+1}|.)$
c_t
a_t
Embedding
$\bar{s}^1$ $\bar{d}^1_1$
$\bar{s}^2$ $\bar{d}^2_1$
$\bar{s}^3$ $\bar{d}^3_1$
$\bar{s}^4$ $\bar{d}^4_1$
$\bar{s}^5$ $\bar{d}^5_1$
$\bar{s}^1$
s^1 d^1_t
s^2 d^2_t
s^3 d^3_t
s^4 d^4_t
s^5 d^5_t

Illustration 3: Attension Mechanism

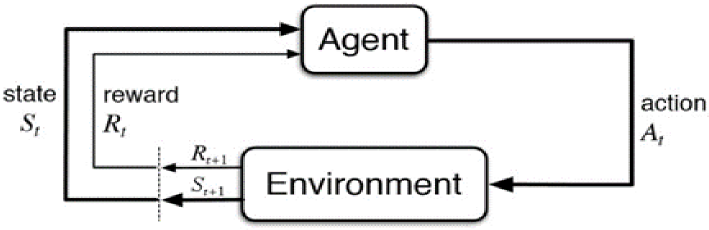


Illustration 4: RL cycle

Attention mechanism is a differential structure for addressing different parts of the input. The attention mechanism employed in our method is illustrated in figure 3.

2.3.1. The Model

In this section, we introduce our model, which is the simplified version of the Pointer Network. We formally define the problem and our proposed framework for generic combinatorial optimization problem with a set of input $X = x^i$, i =1, 2, . . . , M . Some elements of each input is allowed to change between the decoding steps, which is the case in several combinatorial problems such as the VRP. The dynamic elements might be an artifact of the decoding step itself, or they can be imposed by the environment. We formally represent every input x^i by a sequence of tuples $\{x^i_t = (s^i, d^i_t), t = 0,1,\ldots\}$, where s^i and d^i_t are the static and dynamic elements of input, respectively, which can also be a tuples. x^i_t can be viewed as a vector which describes at time *t* the state of input *i*. We will represent the set of all input states at fixed time *t* with X_t . Starting with an arbitrary input in X_0 and pointer y_0 refer to this input. At every decoding time *t*, y_{t+1} points to one of the available X_t , which will determine the input for the next decoding step; and this process goes on and on until a termination condition is satisfied. The termination condition is specific on a particular problem, showing that the sequence generated satisfies the feasibilty constraints. For instance, for the CVRP considered in this work, the terminating condition is satisfied when there is no more demand to be satisfied.

This process will generate sequence of length T, Y = {y_t , t = 0, 1, . . . , T }, with a different sequence-length (probably), when compared with the length of the input sequence *M* . The reason is, the vehicle may have to return to the depot to refill several times. Furthermore, we use Y t to denote the decoded sequence up to time t. Our interest is to find a stochastic policy π that will generates the sequence Y in a way that minimizes a loss objective function while the problem constraints are satisfied. The optimal policy π^* will generate the optimal solution with probability 1. Our goal to to make the optimal gap between π and π^* close to zero. Similar to [17], in order to decompose the probability of generating sequence Y we use the probability chain rule, as follows:

$$P(Y|X_0|) = \prod_{t=0}^{T} P\ (y_{t+1}, X_t), \qquad (2.3.1.1)$$

and

$$X_{t+1} = f(y_{t+1} , X_t), \qquad (2.3.1.2)$$

is a recursive update of the problem representation with f as the state transition function. The right-hand side of equation (2.3.1.2) is computed using the attention mechanism, i.e.,

$$P(y_{t+1} |Y_t , X_t) = softmax\ (g(h_t , X_t)), \qquad (2.3.1.3)$$

where g is an affine function which outputs a vector with input-size, and h t is the RNN state decoder that gives the summaries of previous information on decoded steps, (y_0 , y_1 , . . . , y_t).

3. Results and Discussions

We show the applications of Column Generation, OR-Tools and Reinforcement Learning on Capacitated Vehicle Routing Problem. Our results were compared with best known values for each instances. We used Augerat et. al (set P) data to perform our experiment.

or our experiments, we used HP Elitebook 840 PC, 1.9GHz processor, core i5 with 8GB Memory. We started by showing optimal tour with tables and graphs for Column generation, OR-Tools and comapre them with reinforcement learning.

The optimality gap here is computed as

$$Optimalgap = \frac{Upperbound - Lowerbound}{Lowerbound} \times 100\% \qquad (3.0.1)$$

As we progress, the optimal gap for each technique will given and computed.

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.2636820

3.1. Column Generation Applied on CVRP

The Column Generation (CG) method is an exact method for solving the CVRP and the VRP in general, this technique has been explicitly explained in Section 3.1, and gives an optimal solution to a small-size problem but become inefficient on big-size problem.

Table 3.1 gives the summary of the comparison of this technique's primal and dual problem, since CG work on dual solution of the relaxed master problem, their optimal gap in percentage. *Table 3.1* consists of five columns; Instances, Relaxed Master Problem (RMP), Column Generation (based on dual values), column generation computational time and optimality gap.
These results were obtained using gurobi solver in python [18].

Table 3.1: Column Generation results

Instances	**RMP(Primal)**	**CG value**	**CG time(s)**	**gap**
P-n16-k8	450	450.00	14.72	0.00%
P-n20-k2	220	220.00	38.64	0.00%
P-n22-k2	216	216.00	44.31	0.00%
P-n22-k8	603	603.00	19.66	0.00%

We plots these two solutions; RMP and CG to check for optimality gap. From our previous table, the optimal gap between the two solutions are 0.00%. Hence, the two plots lie on one another. The computational time for the CG in visualized in fig 6.

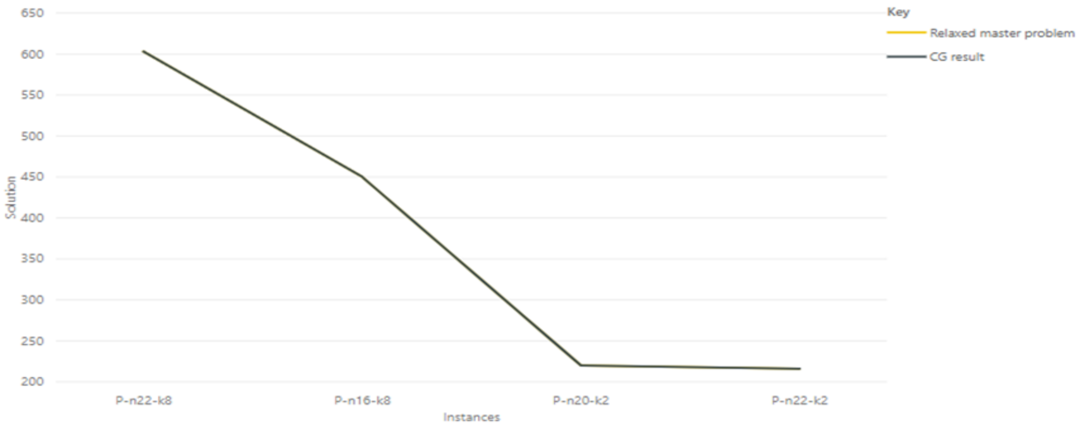


Illustration 5: RMP vs CG plots

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.2636820

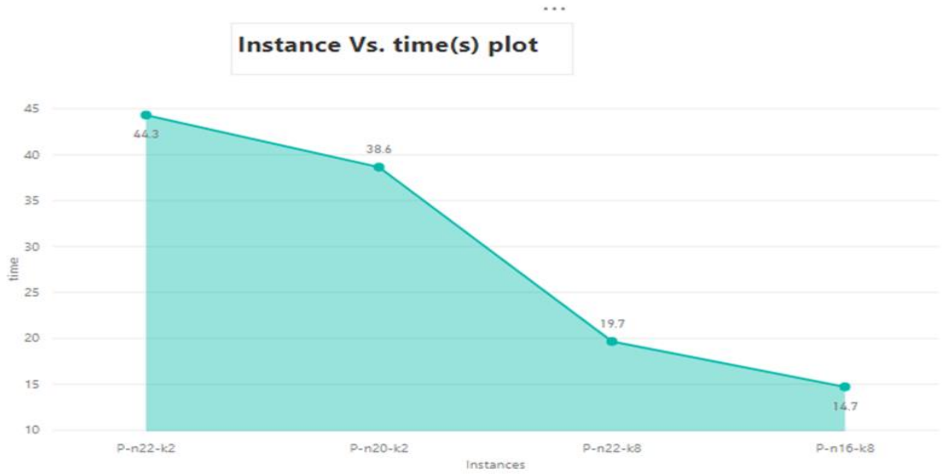


Illustration 6: time plot

Furthermore, we plotted our tours for each instances considered as follows;

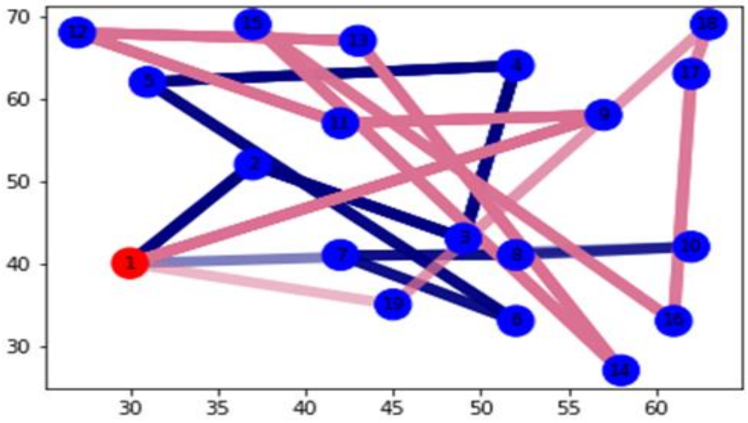


Illustration 7: Graph with 2 routes

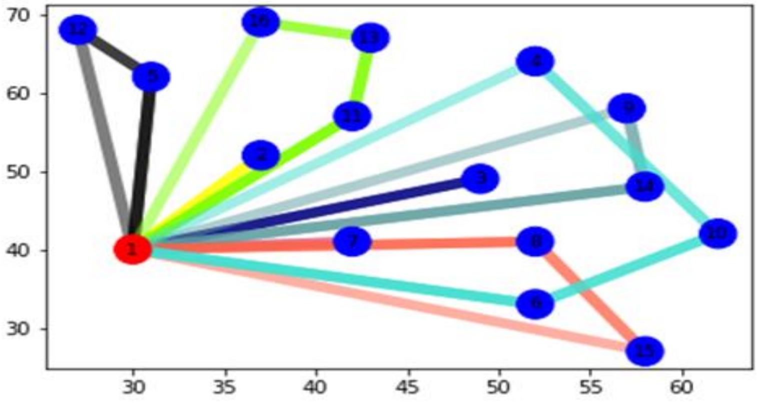


Illustration 7: Graph with 8 routes

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.2636820

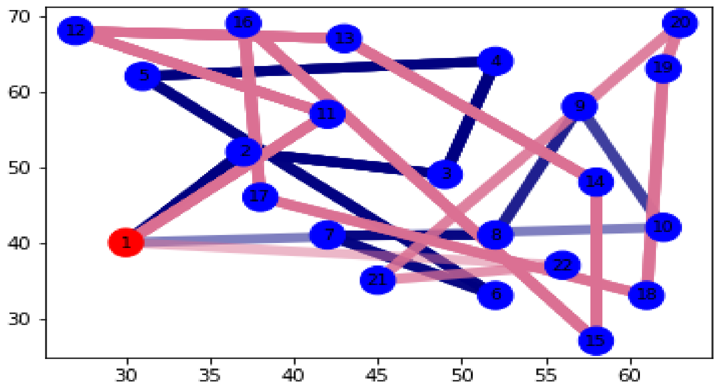


Illustration 8: Graph with 2 routes

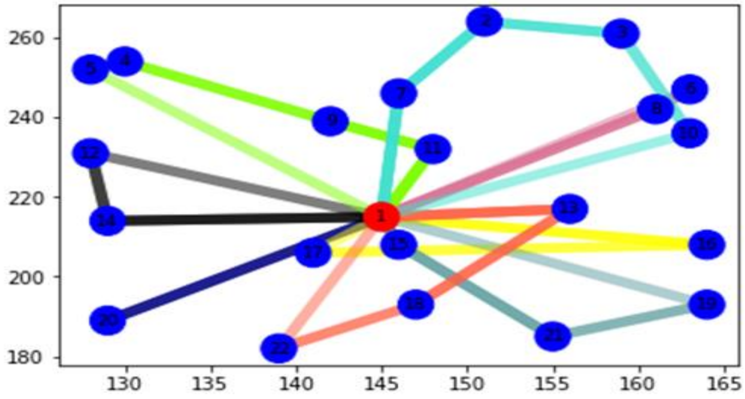


Illustration 9: Graph with 8 route

Table 2: Instance tour

Sample Instances tour using column generation
(1) p-n16-k8 Tour: 1 → 3 → 1 → 7 → 1 → 2 → 1 → 5 → 12 → 1 → 11 → 13 → 16 → 1 → 14 → 9 → 8 → 15 → 1 → 6 → 10 → 4 → 1
(2) p-n20-k2 Tour: 1 → 7 → 6 → 15 → 17 → 10 → 14 → 3 → 11 → 2 → 1 → 5 → 12 → 16 → 13 → 4 → 19 → 18 → 9 → 14 → 8 → 20 → 1
(3) p-n22-k2 tour: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 1 → 11 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 22 → 1
(4) p-n22-k8 tour: 1 → 20 → 1 → 8 → 6 → 1 → 16 → 17 → 1 → 14 → 12 → 1 → 11 → 9 → 4 → 5 → 1 → 15 → 21 → 19 → 1 → 13 → 18 → 22 → 1 → 7 → 2 → 3 → 10 → 1

3.2. OR-Tools applied on CVRP

Operations Research Tools (OR-Tools) is an open source software suitable for solving optimization problems. The OR-Tools is suitable in solving routing problem, constraint programming, flows problem, integer programming and so on [Ope19]. The advantage of this

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.2636820

software to us is that it enables us to find optimal tour and its length for routing problem using python. It computational time is usually very fast compared to other techniques. Its solutions are usually near-optimal when compared with exact method.
We convert the OR-Tools cost formula to euclidean; i.e

$$c = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{3.2.1}$$

Table 3: OR-Tools vs. Best known result

Instances	Best Known	OR-Tools value	OR-Tools time (s)	Optimality gap (%)
A-n32-k5	784*	796	0.04	1.53
P-n16-k8	450*	450	0.50	0.00
P-n20-k2	216*	227	0.02	5.09
P-n22-k2	216*	217	0.02	0.46
P-n22-k8	603*	623	0.51	3.32
P-n40-k5	458*	494	0.10	7.86
P-n50-k7	554*	574	0.08	3.61
P-n70-k10	827*	940	0.17	13.66
P-n101-k4	681*	741	0.35	8.81

From Table 3 above, the or-tools gave a near-optimal solutions. The best known values are better than the or-tools result obtained.

We plot these two solutions; Best known vs. OR-Tools to see the optimality gap as shown in the table, in figure 11. And the instance tour for or-tools is shown in table 4.

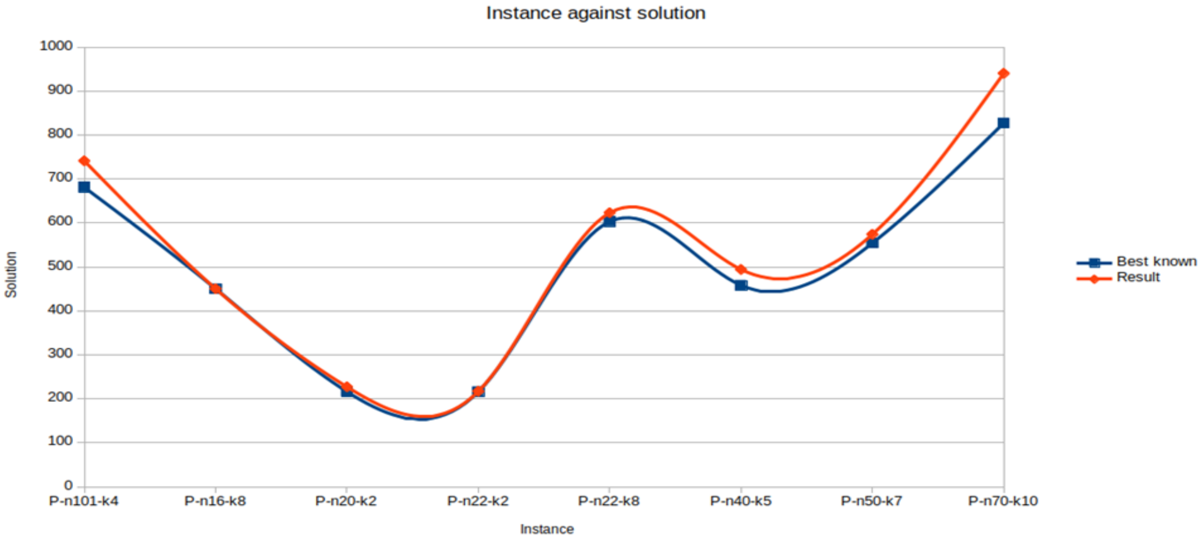


Illustration 10: Solution plots

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.2636820

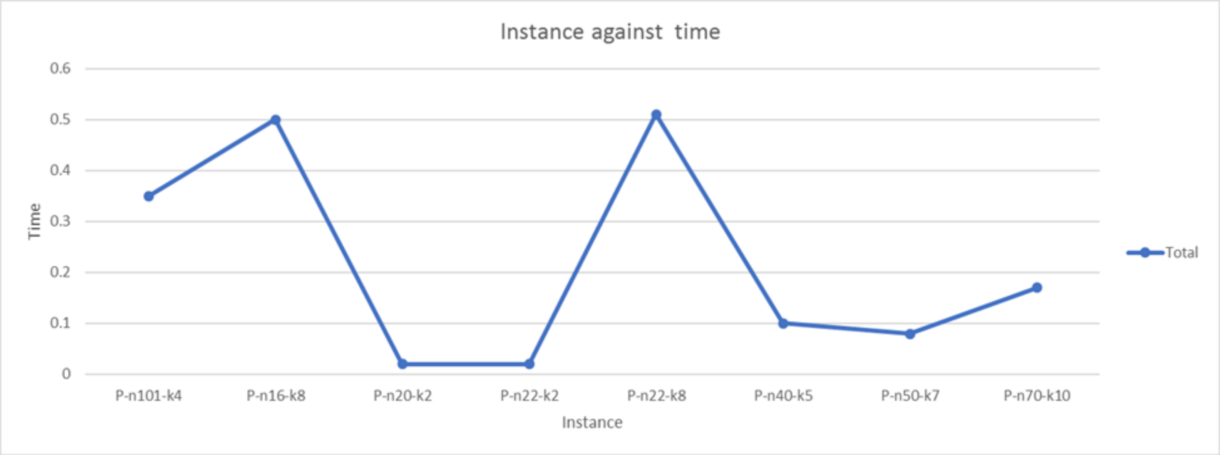


Illustration 11: Time plot

Table 4: Instances tour

Sample Instances tour
(1) *p-n16-k8 Tour:* 0 → 8 → 0 → 5 → 14 → 0 → 10 → 12 → 15 → 0 → 11 → 4 → 0 → 13 → 9 → 7 → 0 → 2 → 0 → 3 → 1 → 6 → 0
(2) *p-n20-k2 Tour:* 0 → 1 → 10 → 4 → 11 → 15 → 12 → 3 → 18 → 17 → 2 → 0 → 19 → 5 → 14 → 16 → 9 → 13 → 8 → 7 → 6 → 0
(3) *p-n22-k2 tour:* 0 → 6 → 2 → 13 → 9 → 17 → 14 → 5 → 21 → 7 → 20 → 0 → 4 → 11 → 15 → 12 → 3 → 19 → 18 → 8 → 10 → 1 → 16 → 0
(4) p-n22-k8 tour:0 → 3 → 1 → 2 → 6 → 0 → 7 → 5 → 0 → 19 → 0 → 17 → 21 → 13 → 0 → 18 → 20 → 14 → 0 → 8 → 4 → 11 → 0 → 12 → 9 → 10 → 0 → 16 → 15 → 0
(5) p-n40-k5 tour:0 → 37 → 17 → 4 → 19 → 13 → 25 → 14 → 6 → 27 → 0 → 24 → 23 → 7 → 26 → 8 → 31 → 28 → 3 → 36 → 35 → 0 → 12 → 18 → 0 → 32 → 2 → 16 → 34 → 21 → 29 → 29 → 20 → 22 → 1 → 0 → 11 → 38 → 9 → 30 → 10 → 39 → 33 → 15 → 5

3.3. CG, OR-Tools and RL with CVRP

Now, we compared the solution of all the three techniques discussed earlier. Table 5 shows our computaional results and their optimal gaps

Table 5: comparison of the three techniques

Instance	**Best Known**	**Column Generation**		**Google's OR-Tools**		**Reinforcement learning**	
		Result(gap)	**time(s)**	**Result(gap)**	**time(s)**	**Result(gap)**	**time(s)**
P-n16-k8	450*	450(0.00%)	14.72	450(0.00%)	0.50	451(0.22%)	0.30
P-n20-k2	216*	216(0.00%)	38.64	227(5.09%)	0.20	220(0.22%)	0.45

ISSN- 2350-0530(O), ISSN- 2394-3629(P)
DOI: 10.5281/zenodo.2636820

P-n22-k2	216*	216(0.00%)	44.31	217(0.46%)	0.20	218(0.93%)	0.55
P-n22-k8	603*	603(0.00%)	19.66	623(3.32%)	0.51	615(2.67%)	1.10

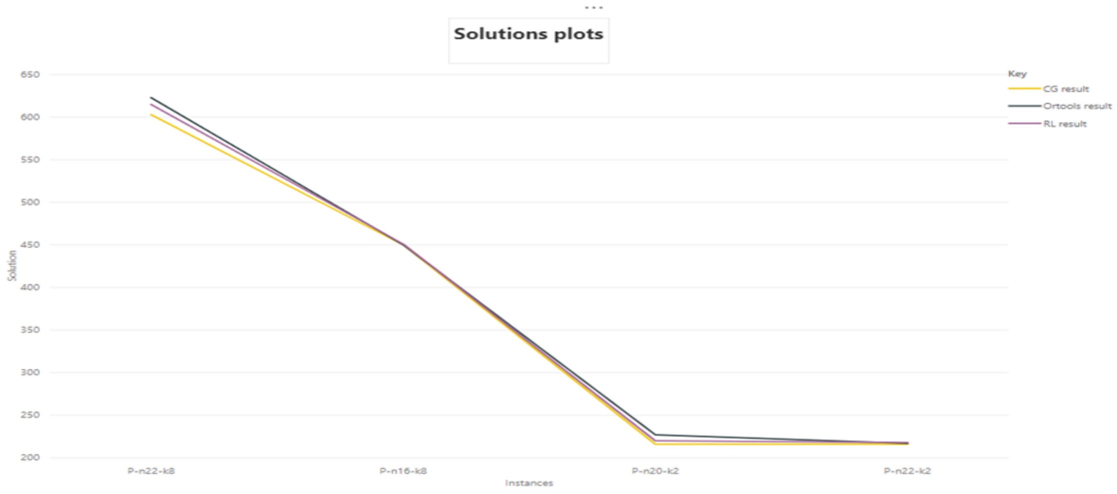


Illustration 13: Solution plots

The solution plots for these three techniques is shown in figure 12 above. Clearly there is gap between these plots and these gaps are shown in table 5 above-mentioned. Below is the computational times for these techniques. We could see the small computational time for or-tools and RL but the CG has high compuation time.

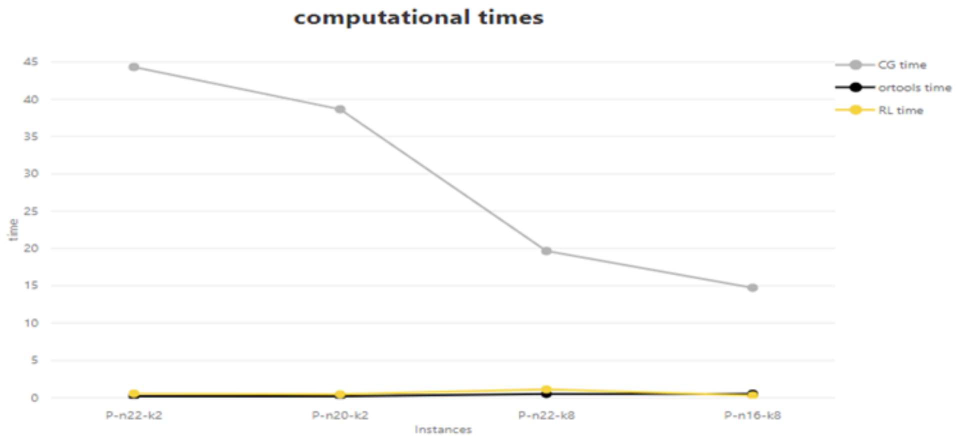


Illustration 12: Computational times

3.4. OR-Tools and RL with CVRP

Google's OR-Tools and RL have been compared with CG previously on small-sized problem. Here, we shall compare these two techniques to solve bigger-size problems. Table 6 shows the path-length and computational time for these techniques and their optimal gap. The "instances" column shows the size of the problems and their vehicle capacity, Q.

DOI: 10.5281/zenodo.2636820

Table 6 below shows result between large-size problems using OR-Tools and RL. Both technique's result are shown with their computational times. Our RL outperforms the google's OR-Tools. Plot 14 shows the solutions and figure 15 shows the computational times for these techniques. The solutions here are scaled down to *[0, 1]* × *[0, 1]* for a better solution.

Table 6: OR-tools Vs. RL results

Instance	**Google's OR-Tools**		**Reinforcement Learning**		**gap**
	path-length	time(s)	path-length	time(s)	
Vrp50, Q=150	31.50	0.29	30.88	02.25	2.01%
Vrp70, Q=135	36.05	0.29	35.74	03.02	0.87%
Vrp100, Q=400	55.62	0.29	55.05	06.23	1.04%

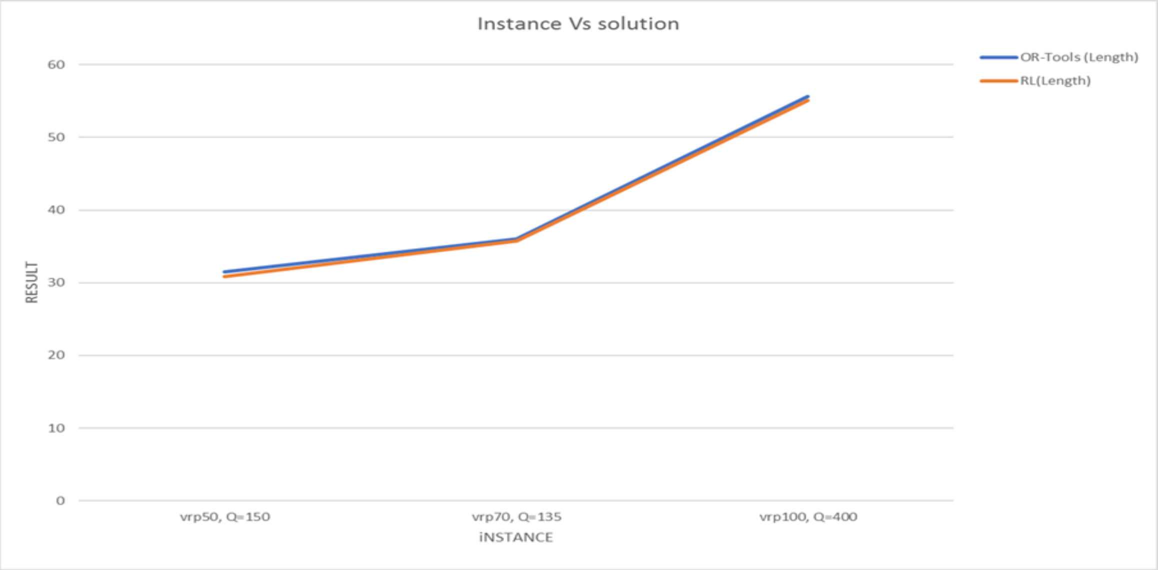
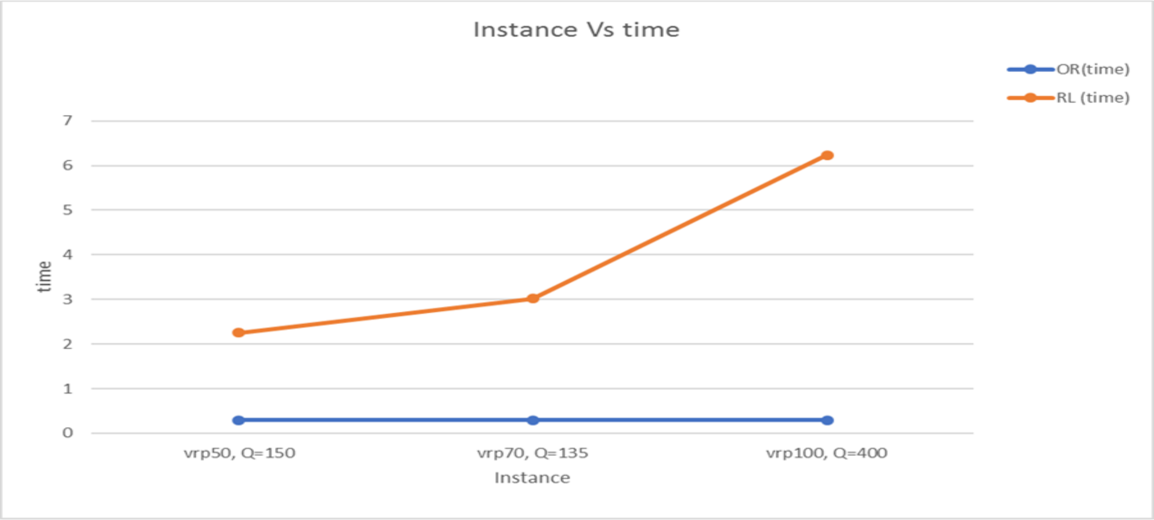


Illustration 16: OR-Tools vs RL plots



We have shown results using various techniques discussed and their tour-length cost, optimal tour, optimality gap and computational time. Each of these techniques is applied on the same data set and the same PC was used to perform these experiments.

4. Conclusions and Recommendations

The Vehicle Routing Problem is an intriguing area of Operational Research which has been studied for several years. The objective is to find efficient routes or paths for transportation of items through a complex network while respecting the capacity constraint of the available vehicles.

The three algorithms discussed here; column generation, or-tools and reinforcement learning algorithms are examined on a small-scale problem. Also, or-tool and reinforcement learning are examined on large scale data.

We end our conclusion with a remark from our aims and objectives. An algorithm that will find a near-optimal solution have been developed. We find the optimal set of routes for a fleet of vehicles delivering goods or services to various locations. In order to achieve this aim, we formulated a mathematical formulation for the Capacitated Vehicle Routing Problem. We went further to solve this formulation with the three techniques; Column generation, Google's Operational Research tool and Reinforcement Learning. We compared the objective values for these techniques with the "best known values" and calculated for the "optimality gap" between this solution, taking the "best known value" as the lower bound. From our experiment, our Reinforcement Learning outperformed the OR-Tools a little bit. Although the computational time for the OR-Tool was faster than the Reinforcement Learning time. These experimental times were very close. The Reinforcement Learning was able to solve a large data set.

5. Future Work

In our future work, a "time window" constraint can be added to this problem. Basically, this problem shall have multiple vehicles with capacity constraint and each location will have a time window. This means that each of the customer/location will require demand at a particular time window $[a_i, b_i]$, where a i is the opening time at a location and a vehicle must arrive on or before a i and b i is the closing time, a vehicle is not allowed to come after b i . Hence, the delivery time, s i to a location must be $a_i \leq s_i \leq b_i$. Customers with similar time window can be merged together as long as the capacity limit of the vehicle is not exceeded.

Acknowledgements

My profound gratitude to my supervisor, Prof. Nassirou Lo and my tutor, Ms. Zoe Hamel towards the success of this research work are well appreciated.

My esteemed appreciation goes to the African Institute for Mathematical Science (AIMS), Senegal for the wonderful opportunity to became a student. The fully funded scholarship, rich international network, the opportunity to work with Development Bank of Rwanda, Kigali, as a data scientist and several other opportunities. All these opportunities made available through AIMS-Senegal, and it is well appreciated.

References

[1] Paolo Toth and Daniele Vigo, An Overview of Vehicle Routing Problems, The Vehicle Routing problem, SIAM, 2002, pp. 1–26.

[2] George B Dantzig and John H Ramser, The Truck Dispatching Problem, Management Science 6 (1959), no. 1, 80–91.

[3] Michael Berliner Pedersen, OB Madsen, and OA Nielsen, Optimization Models and Solution Methods for Intermodal Transportation, Centre for Traffic and Transport, Technical University of Denmark, 2005.

[4] Stefan Ropke, Heuristic and Exact Algorithms for Vehicle Routing Problems, Unpublished PhD thesis, Computer Science Department, University of Copenhagen (2005).

[5] Ricardo Fukasawa, Humberto Longo, Jens Lysgaard, Marcus Poggi de Aragão, Marcelo Reis, Eduardo Uchoa, and Renato F Werneck, Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem, Mathematical Programming 106 (2006), no. 3, 491–511.

[6] Bruce L Golden, Subramanian Raghavan, and Edward A Wasil, The Vehicle Routing Problem: Latest Advances and New Challenges, vol. 43, Springer Science & Business Media, 2008.

[7] Michel Gendreau, Francois Guertin, Jean-Yves Potvin, and René Séguin, Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-ups and Deliveries, Transportation Research Part C: Emerging Technologies 14 (2006), no. 3, 157–174.

[8] GDH Claassen and Th HB Hendriks, An application of special ordered sets to a periodic milk collection problem, European Journal of Operational Research 180 (2007), no. 2,754–769.

[9] Geoff Clarke and John W Wright, Scheduling of vehicles from a central depot to a number of delivery points, Operations research 12 (1964), no. 4, 568–581.

[10] Martin Desrochers, Jacques Desrosiers, and Marius Solomon, A new optimization algorithm for the vehicle routing problem with time windows, Operations research 40 (1992), no. 2,342–354.

[11] Niklas Kohl, Jacques Desrosiers, Oli BG Madsen, Marius M Solomon, and Francois Soumis, 2-path Cuts for the Vehicle Routing Problem with Time Windows, Transportation Science 33 (1999), no. 1, 101–116.

[12] Roberto Baldacci and Aristide Mingozzi, Lower Bounds and an Exact Method for the Capacitated Vehicle Routing Problem, Service Systems and Service Management, 2006 International Conference on, vol. 2, IEEE, 2006, pp. 1536–1540.

[13] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V Snyder, and Martin Takáč, Deep Re-inforcement Learning for Solving the Vehicle Routing Problem, arXiv preprint arXiv:1802.04240(2018).

[14] Frédéric Semet, Paolo Toth, and Daniele Vigo, Chapter 2: Classical Exact Algorithms for the Capacitated Vehicle Routing Problem, Vehicle Routing: Problems, Methods, and Applications, Second Edition, SIAM, 2014, pp. 37–57.

[15] GPT van Lent, Using Column Generation for the Time Dependent Vehicle Routing Problem with Soft Time Windows and Stochastic Travel Times, Master's thesis, 2018.

[16] Inc. google. google's optimization tools, 2019, webpage, https://github.com/google/or-tools, 2019.

[17] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, Sequence to Sequence Learning with Neural Networks, Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[18] Inc. Gurobi Optimization. gurobi optimizer reference manual, 2019, url, http://www.gurobi.com, 2019.

*Corresponding author.
E-mail address: abdullahi.a.ibrahim@ aims-senegal.org