Delivering Pizza with Shadow Prices

Francisco Zenteno Smith
CSE 203B Convex Optimization
fzentenosmith@ucsd.edu

1 Introduction

1.1 Motivation

Pizza delivery can be one of the hardest mathematical problems in engineering. Imagine a pizza shop on a Saturday evening, having received some customer orders that need delivery. The shop has 3 equal vehicles with the capacity to carry a maximum of 4 orders. The manager needs to organize the routes to deliver the orders such that the traveling cost is minimized. How difficult can this be? Figure 1 shows an illustration of this problem.

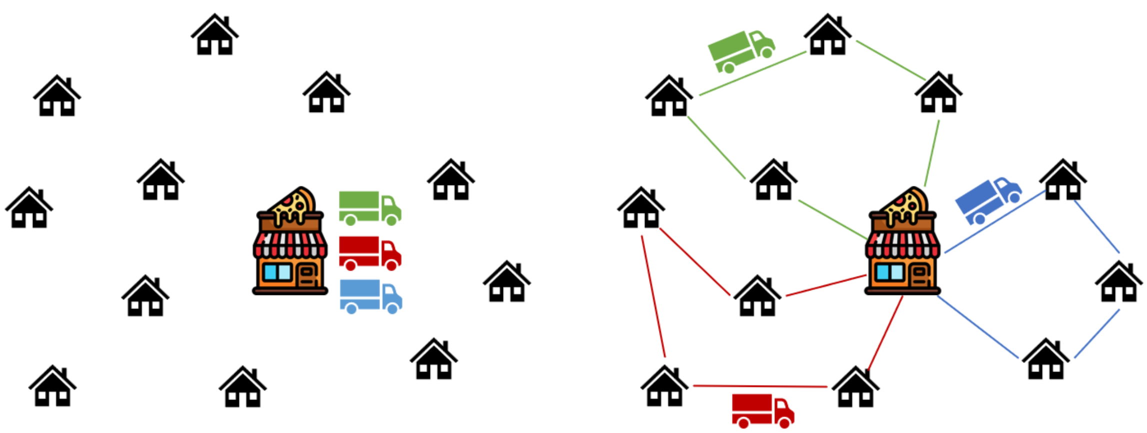


Figure 1: On the left you can see the problem the manager faces. Each black house represents a customer located at a specific address in the city, requiring 1 order of pizza to be delivered. The shop has 3 equal vehicles, depicted in green, red and blue, all of them with the capacity to carry a maximum of 4 orders. On the right you can see a feasible solution of this problem.

Figure 1 shows a feasible solution to this problem, but is this the **optimal solution**? Turns out this is one of the most studied problems in operations research, the so-called Vehicle Routing Problem (VRP). Although easy to state, it is an NP-Hard problem and thus very difficult to solve to optimality. It arises in many industries in different shapes and variations, and it can be a key aspect of the operations of a company. In the Online Food Delivery market, for instance, according to Statista.com, this market is expected to show an annual growth rate of 10.91% in the next 4 years, resulting in a projected market volume of $534 bn by 2028 in the US only [Sta23]. Companies that can provide the best solutions to this problem will have a technological advantage over their competition.

1.2 Previous Work

The vehicle routing problem (VRP) is one of the most studied problems in operations research. Introduced originally by [DR59], it was designed to solve a practical problem of delivering gasoline from a terminal to service stations [Luc19]. According to [Tot+14], a generic verbal definition of the family of vehicle routing problems can be the following:

Given: A set of transportation requests and a fleet of vehicles.

The problem is then to find a plan for the following:

Task: Determine a set of vehicle routes to perform all (or some) transportation requests with the given vehicle fleet at minimum cost; in particular, decide which vehicle handles which requests in which sequence so that all vehicle routes can be feasibly executed.

More than 60 years have elapsed since the introduction of this problem, and thousands of people have dedicated efforts to push the boundaries of knowledge on this problem. There have been approaches proposing exact and heuristic methods, and the family of different VRP has grown in multiple dimensions. As [Tot+14] shows, some of these dimensions are:

- Network structure: for instance, the demand may be associated to nodes and/or arcs.
- Transportation requests: for instance, instead of a vehicle serving customers, there could be multi-modal service or combined shipments of multiple vehicles.
- Constraints: there could be constraints not only of the capacity of the vehicle, but also in route length, different compartments of the vehicle or time windows in the serving of customers.
- Fleet characteristics: the fleet could be heterogeneous, or start from multiple depots.
- Objective function: it could be single-objective or multi-objective (for instance by considering CO_2 emissions).
- Nature of the data: the data could be deterministic or stochastic.

In particular, and for the purpose of this project, we center ourselves in the Capacitated Vehicle Routing Problem (CVRP). In this problem there has been works related to branch-and-bound, lagrangian relaxations, column-generation, branch-and-cut, branch-and-price, heuristics, metaheuristics and a combination between these methods. The reader can get a deep dive in the book by [Tot+14].

1.3 Contributions

The contributions of this project are:

1. To provide a clear explanation and a tutorial of a column-generation approach working on benchmark datasets for the Capacitated Vehicle Routing Problem (CVRP).

2. To be the first step to be taken before learning more complex approaches to solve this important problem in the Operations Research field.

3. To provide an open-source implementation of this method for the community to be shared in academic and industrial environments. All the code and experiments are publicly available in the code repository at `https://github.com/fjzs/pizza_delivery`.

4. To show how duality can be used as a medium to solve real-world integer (and thus, non-convex) optimization problems, such as in the VRP family.

1.4 Organization of the article

This paper is organized as follows: In section 2 we present a formal definition of the problem. In section 3 we present a simple version of the Column Generation method, the main topic of this project. In section 4 we present the experiments executed by the column generation method and a simple heuristic baseline with their results. In section 5 we present the conclusions of this project.

2 Statement of the Problem

The Capacitated Vehicle Routing Problem (CVRP) was formulated in [DR59] as follows: a fleet of k homogeneous vehicles are routed to visit a set of customers, and all the vehicles have a capacity of Q per route. Let $G = (N, A)$ be a directed graph with N being the set of nodes and A being the set of arcs. All the vehicles start from a depot, indicated by node 0, and must return to the depot by the end of the route. The customers are defined by the set $C = N - \{0\}$. Each customer has a demand $d_i > 0 \ \forall i \in C$, and must be served by a single vehicle. On each route a vehicle can only visit a customer once. Every arc $(i, j) \in A$ has a travel cost $c_{ij} \geq 0$. The goal is to minimize the travel costs such that every customer is served.

Desrochers et al [DDS92] showed a formulation called the **Set Partitioning Model** (SPM), where they define:

- R: set of feasible routes for the CVRP.
- a_{ir}: parameter that takes the value 1 if route $r \in R$ serves customer $i \in C$, and 0 otherwise. It satisfies the requirements of a route (not exceding the capacity, visiting each customer once, starting at the depot and finishing at the depot).
- c_r: parameter indicating the cost of route r (which is the sum of the arc's costs of the route).
- x_r: binary variable equalling 1 if route r is used, 0 otherwise.

The formulation is:

$$\min \quad \sum_{r \in R} c_r x_r \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in C} a_{ir} x_r = 1, \quad \forall i \in C \tag{2}$$

$$\sum_{r \in C} x_r = k \tag{3}$$

$$x_r \in \{0,1\}, \quad \forall r \in R \tag{4}$$

The objective function in (1) minimizes the total traveling costs. Constraint (2) ensures every customer is visited once, constraint (3) ensures k vehicles are used (or k routes), and constraint (4) imposes the binary type of the routing choices.

Recall that in a generic optimization problem in the form of $Ax = b$, the rows of the matrix A represent the constraints and the columns the variables. In this context, note that the number of columns (or routes) in the set R is in the order of $\mathcal{O}(n!)$, n being the number of nodes in the graph. Because of this reason, this formulation cannot be solved directly by approaches involving exhaustive column enumeration for even small instances of n.

Nonetheless, a clever observation was made in the early 60's: only a small subset of the columns are going to be selected in the optimal solution. In other words, only a small subset of R is going to be used, in particular for this example that is going to be k elements of R. Thus, in theory, it is possible to get this optimal solution using a small subset of the columns space R as long as the routes in the optimal solution are part of this reduced set. This idea is materialized in a technique called column generation, described below.

3 Column Generation Method

3.1 The Master Problem

Column Generation is an iterative procedure: starting with a subset of columns $R' \subseteq R$, at each iteration this technique decides which columns to add to R'. The procedure stops once the optimal solution is obtained or a stopping criterion is met.

Because we can't enumerate all the potential routes to be used in the SPM, we are going to define a restricted version of this formulation, which is going to be called the 'Restricted Master Problem' (RMP). This problem has the same formulation as the SPM, but it is defined only with a small subset of routes indicated by the set $R' \subseteq R$:

The formulation of the **RMP** is:

$$\min \quad \sum_{r \in R'} c_r x_r \tag{5}$$

$$\text{s.t.} \quad \sum_{i \in C} a_{ir} x_r = 1, \quad \forall i \in C \tag{6}$$

$$x_r \in \{0,1\}, \quad \forall r \in R' \tag{7}$$

The only difference between this formulation (RMP) and the original one (SPM) is the number of routes (columns) considered. In the RMP we are considering R', whereas in the SMP we are considering all the routes possible R ([DL05]).

Column generation methods leverage the information provided by duality to incorporate new columns. In Figure 2 a summary of the primal and dual formulation is shown. Recall from linear programming theory that a reduced-cost variable is calculated by $c - A^T u$ (where u are the dual variables associated with the constraints in A). If this reduced cost is negative, then this variable can enter the base in the primal formulation and improve the value of the objective function.

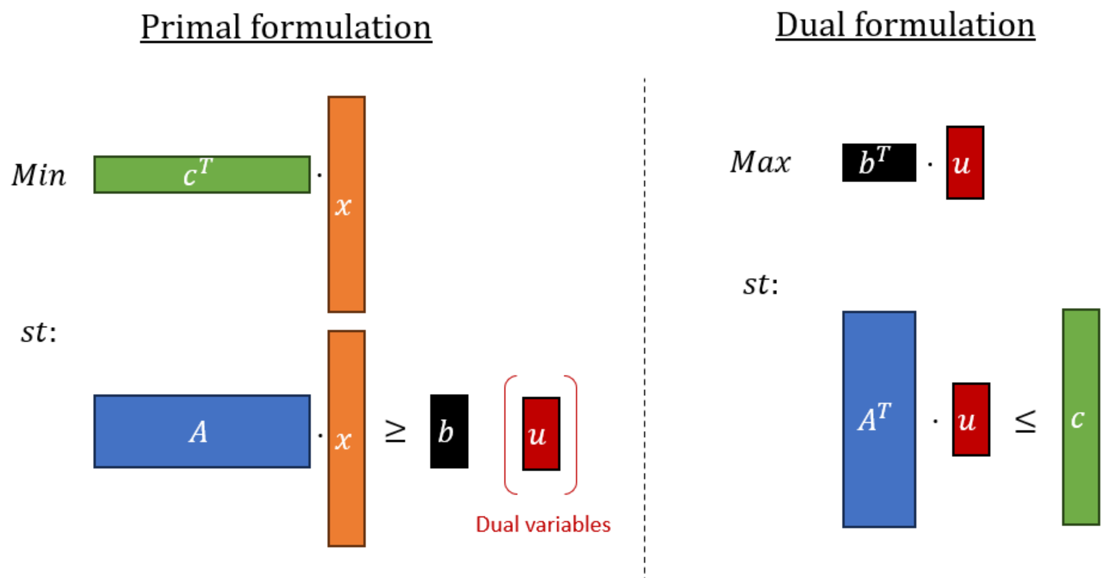


Figure 2: In this figure you can see the primal and dual formulation of a linear programming problem. The heights and widths of the rectangles represent the shapes of these elements in the problem. For instance, the height of A represents the number of constraints and its width the number of variables (or columns) in the primal formulation.

In the context of the CVRP, the matrix A represents contraint (6) of the RMP, the vector b is a vector of ones and u are the shadow prices associated with each customer $c \in C$. Nonetheless, this theory can't be applied to the RMP or SPM because these are integer programming problems. To overcome this, column generation solves the linear version of the RMP, which we will denote **L-RMP**:

$$\min \quad \sum_{r \in R'} c_r x_r \tag{8}$$

$$\text{s.t.} \quad \sum_{i \in C} a_{ir} x_r \geq 1, \quad \forall i \in C \tag{9}$$

$$0 \leq x_r \leq 1, \quad \forall r \in R' \tag{10}$$

The L-RMP differs from the RMP in constraint (9) which uses a $\geq$ symbol to get positive shadow prices u_i for every customer. It also relaxes the binary type of x_r to be continuous as shown in constraint (10).

To understand how shadow prices can help the L-RMP, let's take a look at the dual formulation of the L-RMP, the **Dual-L-RMP**:

$$\max \quad \sum_{i \in C} u_i \tag{11}$$

$$\text{s.t.} \quad \sum_{i \in C} a_{ir} u_i \leq c_r, \quad \forall r \in R' \tag{12}$$

$$u_i \geq 0, \quad \forall i \in C \tag{13}$$

In the Dual-L-RMP the variable u_i are the shadow prices of the primal formulation, which represent the marginal cost of serving customer i. Constraint (12) ensures that the sum of the marginal costs of a route $r \in R'$ is less than the cost of that route. The column generation process is equivalent to generating lazy cuts in the dual formulation. At each iteration, only a subset of the constraints (those routes in R') are included in the dual formulation. If we solve this problem and find a violated constraint associated with a route r^*, then it implies that the reduced cost of that route is negative, and thus it will improve the bound in the primal formulation. If there is no such route, then the primal formulation was been solved to optimality. Finding this violated constraint is what is called the Pricing Problem [BJC22].

3.2 The Pricing Problem

The Pricing Problem is an optimization problem that uses the shadow prices from the L-RMP to generate reduced-cost variables, hence its name. Usually, the benefit of decomposing the RMP into its linear relaxation L-RMP and a Pricing Problem arises from the fact that the Pricing Problem can be solved in polynomial or pseudo-polynomial time, resembling this approach to a divide-and-conquer technique.

[DL05] indicates that the Pricing Problem of the CVRP decomposes into an Elementary Shortest Path Problem and Capacity Constraints (ESPCC). Elementary in this case means that each customer can appear at most once in the shortest path. The description of this problem is shown in the next paragraph.

Consider $N = \{0, 1, ..., n+1\}$ a set of nodes where node 0 is the source and node $n+1$ is the target of a path to find. Consider the set of arcs $A = \{(i,j) | i \in N - \{n+1\}, j \in N - \{0\}\}$. In other words, there are no arcs arriving at the source node, and there are no arcs starting from the target node. Each arc $(i,j) \in A$ has a cost $\bar{c}_{ij} \in \mathbb{R}$ (so it can be negative). There is a capacity given by q and the consumption of each node is given by $d_i \in N : d_0 = d_{n+1} = 0$. The goal is to minimize the path cost between the source and the target node. Because the ESPCC is solved as a subproblem of the CVPR, $\bar{c}_{ij} = c_{ij} - u_i$ represents the reduced cost of a route in the L-RMP, where u_i are the shadow prices associated to constraint (9) in the L-RMP. The formulation of the **ESPCC** is:

$$\max \quad \sum_{(i,j)\in A} \bar{c}_{ij} x_{ij} \tag{14}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A} d_i x_{ij} \leq q \tag{15}$$

$$\sum_{j\in N} x_{0j} = 1 \tag{16}$$

$$\sum_{i\in N} x_{ih} = \sum_{j\in N} x_{hj} \quad \forall h \in N - \{0, n+1\} \tag{17}$$

$$\sum_{i\in N} x_{i,n+1} = 1 \tag{18}$$

$$\sum_{(i,j)\in A,(i,j)\in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \tag{19}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \tag{20}$$

Constraint (15) is the capacity constraint, (16) - (18) are flow balance constraints to ensure a path solution from the source to the target. Constraints (19) are subtour elimination constraints and (20) ensures integrality.

This is a well-known NP-Hard problem, and thus unfortunately it is not easy to solve. Nonetheless, the column generation method only requires solutions to this problem with negative reduced costs, not necessarily the optimal ones. Fortunately, the Operations Research community has worked intensively on this problem, and has developed several techniques to avoid solving this problem exactly to instead provide quick and heuristic solutions, which we will use.

In particular, we will use the python package CSPY [Tor20], a library that implements different exact and metaheuristic algorithms to solve this problem, such as [Til+17], [DG20] or [MMS17].

3.3 Summary

The column generation method is summarized in Figure (3). Recall that:

- MP stands for the Master Problem, which is an integer optimization problem.
- RMP stands for Restricted Master Problem, which is the MP with a restricted set of columns. This is also an integer optimization problem.
- L-RMP stands for Linearized Restricted Master Problem, which is the linear version of the RMP.

When the column generation method ends we will have a set of potential routes (columns) in R'. Then we can use branch & bound to solve the RMP. Nonetheless, this method does not guarantee an optimal solution in the RMP, only in the L-RMP. An optimal solution can be obtained by applying a procedure called branch-price-and-cut, which is out of the scope of this project.

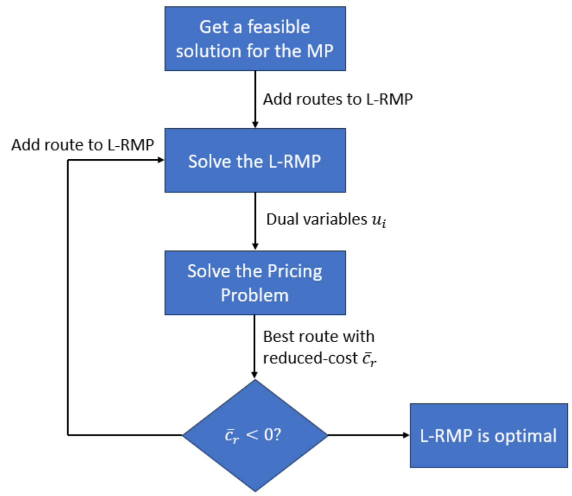


Figure 3: The column generation method is summarized in this diagram. The pricing problem has to be solved until there are no more negative reduced-cost routes.

3.4 Example

Here I show an example of this method working in a small instance of 4 nodes, shown in Figure (4). Coincidently, this example is also shown in [Fei10].

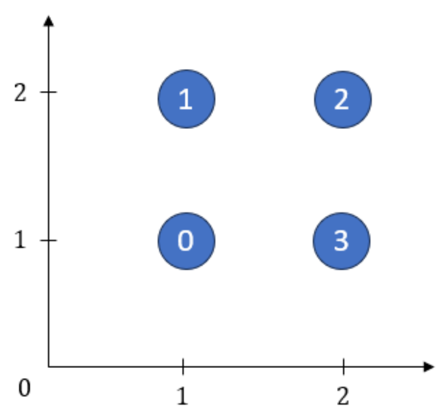


Figure 4: This instance is composed of 4 nodes, $N = \{0, 1, 2, 3\}$. and 3 customers (all nodes except 0). The nodes are placed on a 2D Euclidean plane, so all the vertical and horizontal distances are 1, and the diagonal distances are $\sqrt{2} \approx 1.4$. Assume the capacity of each vehicle is $q = 3$.

So if we wanted to formulate the MP, the columns (variables, or routes) of the problem would be the ones shown in Figure (5).

Route id	Route	Cost
1	[0, 1, 0]	2
2	[0, 2, 0]	2.8
3	[0, 3, 0]	2
4	[0, 1, 2, 0]	3.4
5	[0, 2, 1, 0]	3.4
6	[0, 1, 3, 0]	3.4
7	[0, 3, 1, 0]	3.4
8	[0, 2, 3, 0]	3.4
9	[0, 3, 2, 0]	3.4
10	[0, 1, 2, 3, 0]	4
11	[0, 1, 3, 2, 0]	4.8
12	[0, 2, 1, 3, 0]	4.8
13	[0, 2, 3, 1, 0]	4.8
14	[0, 3, 1, 2, 0]	4.8
15	[0, 3, 2, 1, 0]	4

Figure 5: All possible columns of the MP.

The objective function is:

$Min\ \sum_r x_r c_r = 2x_1 + 2.8x_2 + 2x_3 + 3.4x_4 + 3.4x_5 + 3.4x_6 + 3.4x_7 + 3.4x_8 + 3.4x_9 + 4x_{10} + 4.8x_{11} + 4.8x_{12} + 4.8x_{13} + 4.8x_{14} + 4x_{15}$.

And the constraints are the ones shown in Figure (6).

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15			
1			1	1	1	1			1	1	1	1	1	1	≥	1	(Client 1)
	1		1	1			1	1	1	1	1	1	1	1	≥	1	(Client 2)
		1			1	1	1	1	1	1	1	1	1	1	≥	1	(Client 3)

Figure 6: Constraints of the MP, the implementation of constraint (2) in the SPM. We also need to add binary constraints, which are not shown here for simplicity.

Now, let's assume that we can't enumerate all the columns in advance (because the number is in the order of $O(n!)$. The first step in the algorithm is to get a feasible solution for the MP.

Getting an initial solution

Let's define the trivial solution of sending one vehicle to each customer once. Therefore, the restricted routes set is $R' = \{1, 2, 3\}$ which form the initial solution, shown in Figure (7).

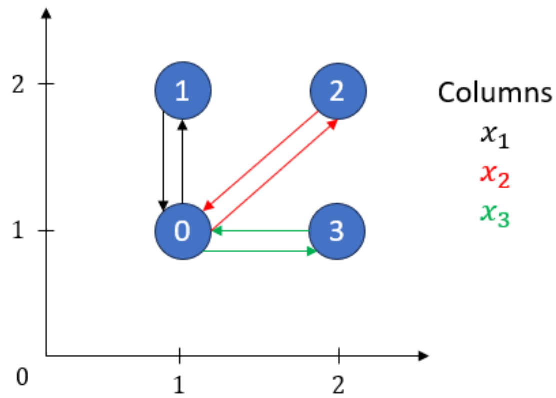


Figure 7: Initial solution of the Restricted Master Problem. There are only 3 columns possible, and all of them are in the feasible solution.

Solve the L-RMP iteration # 1
The current L-RMP would look like this:

$$\min \sum_{r \in R'} x_r c_r = 2x_1 + 2.8x_2 + 2x_3$$

s.t.

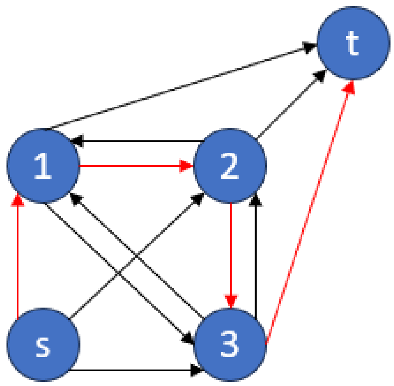
$$x_1 \geq 1$$
$$x_2 \geq 1$$
$$x_3 \geq 1$$
$$0 \leq x_r \leq 1, \ \forall r \in R'$$

After solving this problem the optimal objective value is, of course, $Z_{L-RMP} = 2 + 2.8 + 2 = 6.8$, and the dual variables obtained by solving this instance are: $\{u_1 = 2, \ u_2 = 2.8, \ u_3 = 2\}$. With the duals we can proceed to the pricing problem.

Solve the Pricing Problem iteration # 1
Recall that the pricing problem tries to find a reduced-cost route, where each arc has a transformed cost of $\bar{c}_{ij} = c_{ij} - u_i$. The instance of this first pricing problem is shown in Figure (8). The optimal solution of this problem is denoted in the red arcs, and has an objective value of -2.8. Because this is a negative reduced-cost route, it now enters the set R' in the L-RMP.

Solve the L-RMP iteration # 2
After adding the route $x_{10} = [0, 1, 2, 3, 0]$[1] with cost $c_{10} = 4$, the current L-RMP looks like this:

[1]We are going to denote it with the id #10 just to maintain consistency with the set R, but in practice the id would be 4.



Arc	Reduced cost	Value
(s, 1)	1 - u_s	1
(s, 2)	1.4 - u_s	1.4
(s, 3)	1 - u_s	1
(1, 2)	1 - u_1	-1
(1, 3)	1.4 - u_1	-0.6
(1, t)	1 - u_1	-1
(2, 1)	1 - u_2	-1.8
(2, 3)	1 - u_2	-1.8
(2, t)	1.4 - u_2	-1.4
(3, 1)	1.4 - u_3	-0.6
(3, 2)	1 - u_3	-1
(3, t)	1 - u_3	-1

Duals	Value
u_s	0
u_1	2
u_2	2.8
u_3	2
u_t	0

Figure 8: Pricing Problem iteration #1. The network is composed of a source node denoted by s and a target node denoted by t (the depot nodes). Each arc has a modified cost given the dual variables. The optimal solution is shown in the red arcs.

$$\min \sum_{r \in R'} x_r c_r = 2x_1 + 2.8x_2 + 2x_3 + 4x_{10}$$

s.t:

$$x_1 + x_{10} \geq 1$$

$$x_2 + x_{10} \geq 1$$

$$x_3 + x_{10} \geq 1$$

$$0 \leq x_r \leq 1, \ \forall r \in R'$$

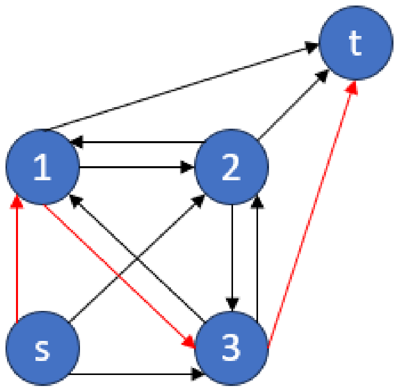
After solving this problem the optimal solution only has x_{10} in the base, so the objective value is $Z_{L-RMP} = 4$, and the dual variables obtained by solving this instance are: $\{u_1 = 2, \ u_2 = 0, \ u_3 = 2\}$. With the duals we can proceed to the pricing problem.

Solve the Pricing Problem iteration # 2

The instance of this second pricing problem is shown in Figure (9). The optimal solution of this problem is denoted in the red arcs, and has an objective value of -0.6. Because this is a negative reduced-cost route, it now enters the set R' in the L-RMP.

Solve the L-RMP iteration # 3

After adding the route $x_6 = [0, 1, 3, 0]$ with cost $c_6 = 3.4$, the current L-RMP looks like this:



Arc	Reduced cost	Value
(s, 1)	1 - u_s	1
(s, 2)	1.4 - u_s	1.4
(s, 3)	1 - u_s	1
(1, 2)	1 - u_1	-1
(1, 3)	1.4 - u_1	-0.6
(1, t)	1 - u_1	-1
(2, 1)	1 - u_2	1
(2, 3)	1 - u_2	1
(2, t)	1.4 - u_2	1.4
(3, 1)	1.4 - u_3	-0.6
(3, 2)	1 - u_3	-1
(3, t)	1 - u_3	-1

Duals	Value
u_s	0
u_1	2
u_2	0
u_3	2
u_t	0

Figure 9: Pricing Problem iteration #2. The optimal solution is shown in the red arcs.

$$\min \sum_{r \in R'} x_r c_r = 2x_1 + 2.8x_2 + 2x_3 + 3.4x_6 + 4x_{10}$$

s.t:

$$x_1 + x_6 + x_{10} \geq 1$$
$$x_2 + x_{10} \geq 1$$
$$x_3 + x_6 + x_{10} \geq 1$$
$$0 \leq x_r \leq 1, \ \forall r \in R'$$

After solving this problem the optimal solution only has x_{10} in the base, so the objective value is $Z_{L-RMP} = 4$, and the dual variables obtained by solving this instance are: $\{u_1 = 1.4, \ u_2 = 0.6, \ u_3 = 2\}$. With the duals we can proceed to the pricing problem.

Solve the Pricing Problem iteration # 3

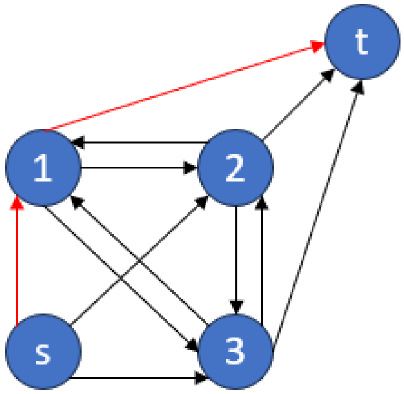
The instance of this third pricing problem is shown in Figure (10). The optimal solution of this problem is denoted in the red arcs, and has an objective value of 0. Because this is not a negative reduced-cost route, the L-RMP has now been solved to optimality and the loop ends here.

A couple of things to note in this example:

1. The optimal solution in the L-RMP was obtained only with 5 out of the 15 columns.

2. Although we found the optimal solution to the MP, this was only a coincidence and due to the simplicity of this example.

The optimal solution to the L-RMP is shown in Figure (11)



Arc	Reduced cost	Value
(s, 1)	1 - u_s	1
(s, 2)	1.4 - u_s	1.4
(s, 3)	1 - u_s	1
(1, 2)	1 - u_1	-0.4
(1, 3)	1.4 - u_1	0
(1, t)	1 - u_1	-0.4
(2, 1)	1 - u_2	0.4
(2, 3)	1 - u_2	0.4
(2, t)	1.4 - u_2	0.8
(3, 1)	1.4 - u_3	-0.6
(3, 2)	1 - u_3	-1
(3, t)	1 - u_3	-1

Duals	Value
u_s	0
u_1	1.4
u_2	0.6
u_3	2
u_t	0

Figure 10: Pricing Problem iteration #3. The optimal solution is shown in the red arcs and has a value of 0.

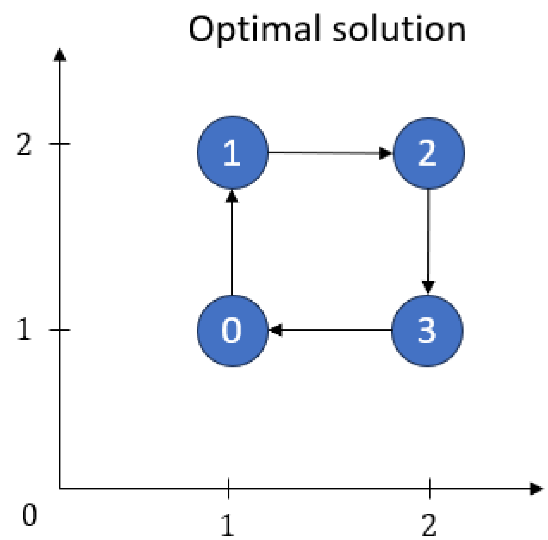


Figure 11: Optimal solution of the L-RMP. Coincidentally it is also the optimal solution of the MP. It is only composed of the column $x_{10} = [0, 1, 2, 3, 0]$ with cost $c_{10} = 4$.

4 Experiments

4.1 Instances

We are going to analyze the performance of a simple column generation method in 3 known instances used to benchmark the CVRP. The instances are named 'E-n22-k4' [CE69], 'A-n33-k6' [Aug95] and

'CMT1' [**CMT**]. The details of these instances can be found on this website: `http://vrp.atd-lab.inf.puc-rio.br/index.php/en/` and a summary of the characteristics are shown in Table 1.

Characteristics	E-n22-k4	A-n33-k6	CMT1
Number of customers	21	32	50
Minimum number of vehicles	4	6	5
Capacity	6000	100	160
Optimal value	375	742	524
Demand range	[1, 2500]	[1,66]	[1,41]

Table 1: These 3 instances are ordered in increasing size (note the number of customers), and thus they have increasing difficulty.

Diagrams or maps of the instances are shown in Figures 12, 13 and 14. In all these instances the distances between nodes are computed as a 2D Euclidean distance.

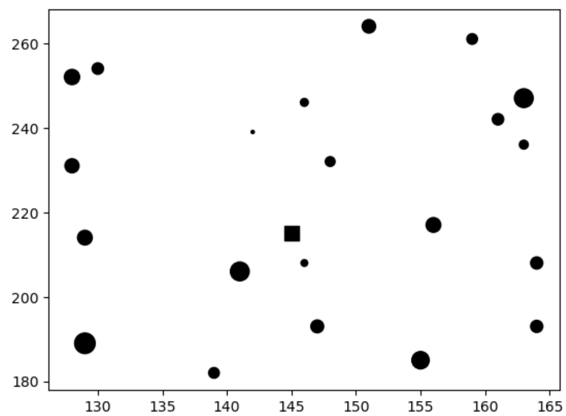


Figure 12: E-n22-k4 instance. The square represents the depot and the circles are the customers. The size of the circle is proportional to the demand of the customer.

4.2 Results

The method column generation was evaluated against a simple heuristic. This heuristic has 2 parts: (1) an initializer, which has the responsibility of providing a feasible solution, and (2) an improver, which has the responsibility of improving the initial solution. The initializer picked was the well-known Clarke & Wright heuristic [CW64], and the improver was the 2-OPT heuristic [Lin65]. The objective function results are shown in Table 2 and the solutions are shown in Figures 15, 16 and 17.

Results show that the column generation method beats the simple heuristic in all the instances. Coincidentally in the instance E-n22-k4 the column generation method achieves the optimal solution, but note that this is not usually the case for this method (in order to achieve better results more complex methods such as branch-and-price are required).

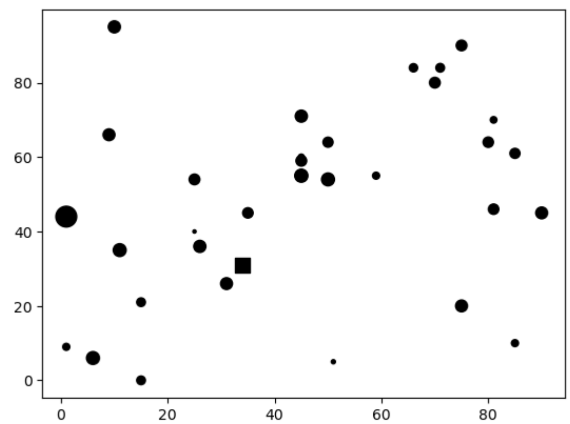


Figure 13: A-n33-k6 instance. The square represents the depot and the circles are the customers. The size of the circle is proportional to the demand of the customer.

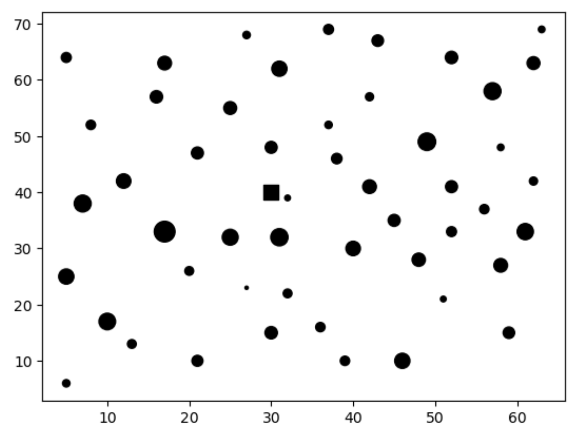


Figure 14: CMT1 instance. The square represents the depot and the circles are the customers. The size of the circle is proportional to the demand of the customer.

Although not shown explicitly, this heuristic runs 10x faster than the column generation method, and thus it could be useful for applications in which speed is crucial and where it could be traded-off for optimality.

Finally, a video showing the evolution of the objective function in the RMP and the solution is displayed in the following links. It might be useful for the reader to see what the method is doing.

- Instance E-n22-k4: `https://www.youtube.com/watch?v=V23W4LGLY-M`
- Instance A-n33-k6: `https://www.youtube.com/watch?v=BSFrKqD6E44`
- Instance CMT1: `https://www.youtube.com/watch?v=xKCxvxl09ms`

Instance	Heuristic	Column Generation	Optimal Value
E-n22-k4	387 (3.2%)	375 (0%)	375
A-n33-k6	759 (2.3%)	750 (1.1%)	742
CMT1	604 (15.3%)	583 (11.3%)	524

Table 2: Objective function values achieved by both methods in every instance. The number between parenthesis is the optimality gap of that method with respect to the optimal value of the instance. First thing to notice is that the column generation approach beats the heuristic in all the instances. Only in the first instance, and only by coincidence, the column generation method arrives to the optimal value. As the size of the instance increases, the column generation method optimality gap increases as well.

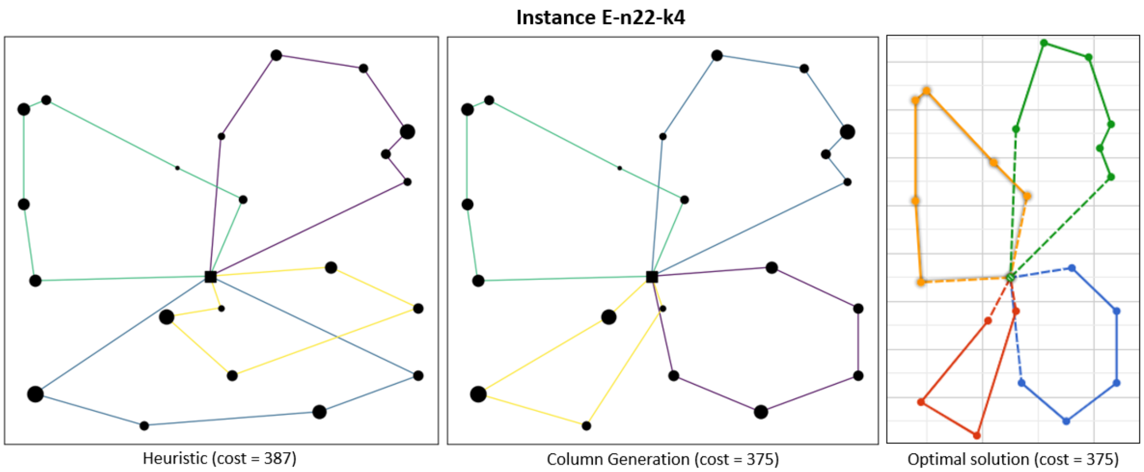


Figure 15: In this instance the column generation method achieves an optimal value. Although both the heuristic and the column generation method use 4 vehicles, just as the optimal solution, the heuristic fails to find the optimal value.

5 Conclusion

The conclusions of this project are:

- We showed step-by-step how column generation method works in the context of one of the simplest version of the VRP family of problems, the Capacitated Vehicle Routing Problem (CVRP).
- We showed that this method is a component of more complex state-of-the-art approaches to solve the CVRP. In particular, the column generation method showed in a key component in a complex method called branch-and-price, used to solve one of the most difficult instances of the CVRP.
- We also showed that heuristic approaches can also solve this problem, and that even though they not guarantee optimal solutions, they could be much faster and useful for business applications.

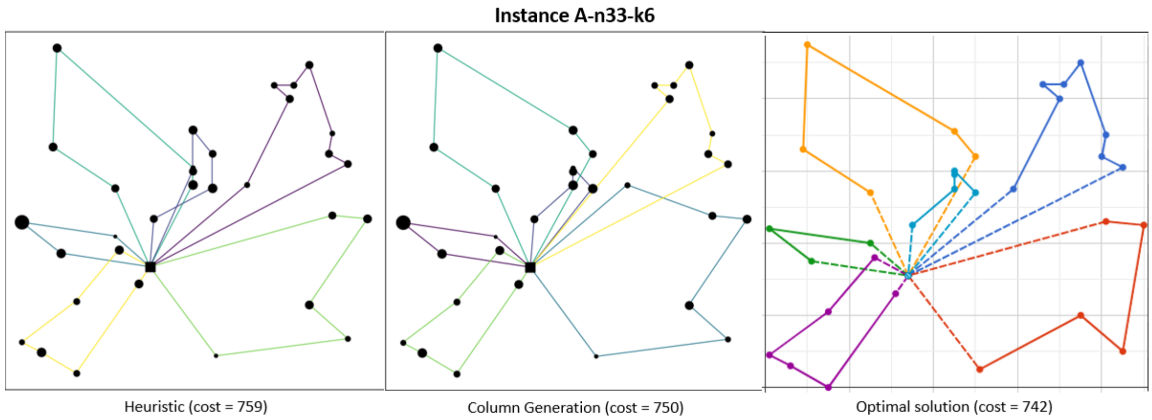


Figure 16: In this instance both the heuristic and column generation method use 6 vehicles, but the later method finds a better solution, with some similar routes as in the optimal solution.

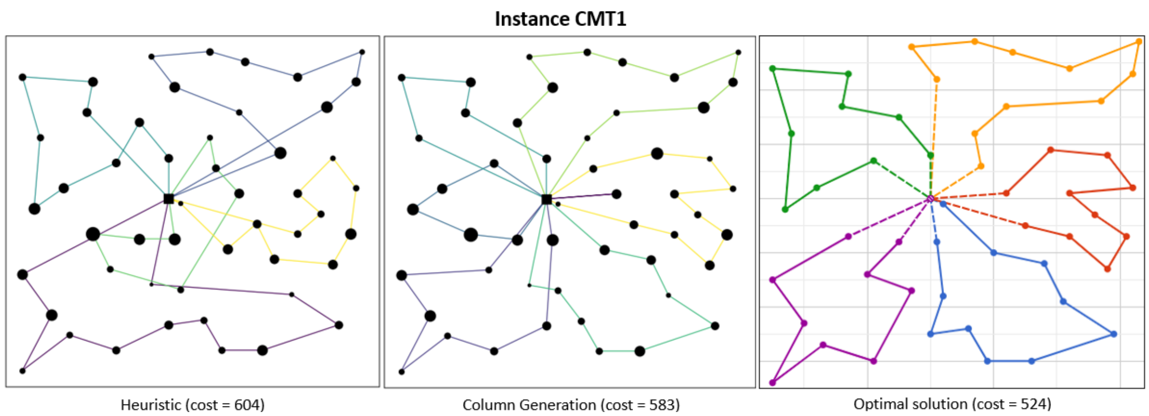


Figure 17: This is the hardest instance, and although the heuristic uses 5 vehicles and column generation uses 7, the later finds a better solution. In bigger instances we can see how this simple column generation method starts to be insufficient for finding close to optimal solutions (in this case it was more than 10% in optimality gap).

- We provided an open-source implementation of this tutorial, which could be used in both academic or industrial environments.
- Finally, we showed how shadow prices and duality theory can be leveraged to solve this complex integer and non-convex problem.

References

[DR59] George B. Dantzig and John H. Ramser. "The Truck Dispatching Problem". In: *Management Science* 6.1 (1959), pp. 80–91.

[CW64] G. Clarke and J. Wright. “ScheduHng of vehicles from a central depot to a number of delivery points”. In: *Operations Research* 12 (1964), pp. 568–581.

[Lin65] S. Lin. “Computer solutions of the traveling salesman problem”. In: *Bell System Technical Journal* 44 (1965), pp. 2245–2269.

[CE69] N. Christofides and S Eilon. “An algorithm for the vehicle routing dispatching problem”. In: *Operations Research Quaterly* 20 (1969), pp. 309–318.

[DDS92] Martin Desrochers, Jacques Desrosiers, and Marius M Solomon. “A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows”. In: *Operations Research* 40.2 (1992), pp. 342–354.

[Aug95] Philippe Augerat. “Approche polyèdrale du problème de tournées de véhicules”. Theses. Institut National Polytechnique de Grenoble - INPG, June 1995. URL: `https://theses.hal.science/tel-00005026`.

[DL05] Jacques Desrosiers and Marco E. Lübbecke. *Column Generation*. Gerad 25th Anniversary Series. New York, NY: Springer, 2005. DOI: `10.1007/b130422`.

[Fei10] Dominique Feillet. “A tutorial on column generation and branch-and-price for vehicle routing problems”. In: *4OR* 8 (2010), pp. 407–424. URL: `https://api.semanticscholar.org/CorpusID:9031916`.

[Tot+14] Paolo Toth et al. *Vehicle Routing: Problems, Methods, and Applications, Second Edition*. USA: Society for Industrial and Applied Mathematics, 2014. ISBN: 1611973589.

[MMS17] Yannis Marinakis, Athanasios Migdalas, and Angelo Sifaleras. “A hybrid Particle Swarm Optimization – Variable Neighborhood Search algorithm for Constrained Shortest Path problems”. In: *European Journal of Operational Research* 261.3 (2017), pp. 819–834. ISSN: 0377-2217. DOI: `https://doi.org/10.1016/j.ejor.2017.03.031`. URL: `https://www.sciencedirect.com/science/article/pii/S0377221717302357`.

[Til+17] Christian Tilk et al. “Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster”. In: *European Journal of Operational Research* 261.2 (2017), pp. 530–539. ISSN: 0377-2217. DOI: `https://doi.org/10.1016/j.ejor.2017.03.017`. URL: `https://www.sciencedirect.com/science/article/pii/S0377221717302035`.

[Luc19] Guy Desaulniers Luciano Costa Claudio Contardo. “Exact Branch-Price-and-Cut Algorithms for Vehicle Routing”. In: *Transportation Science* 53 (2019), pp. 946–985.

[DG20] Paola Festa Daniele Ferone and Francesca Guerriero. “An efficient exact approach for the constrained shortest path tour problem”. In: *Optimization Methods and Software* 35.1 (2020), pp. 1–20. DOI: `10.1080/10556788.2018.1548015`. eprint: `https://doi.org/10.1080/10556788.2018.1548015`. URL: `https://doi.org/10.1080/10556788.2018.1548015`.

[Tor20] David Torres Sanchez. “cspy: A Python package with a collection of algorithms for the (Resource) Constrained Shortest Path problem”. In: *Journal of Open Source Software* 5.49 (2020), p. 1655. DOI: `10.21105/joss.01655`. URL: `https://doi.org/10.21105/joss.01655`.

[BJC22] Behrouz Babaki, Sanjay Dominik Jena, and Laurent Charlin. "Neural Column Generation for Capacitated Vehicle Routing". In: *AAAI-22 Workshop on Machine Learning for Operations Research (ML4OR)*. 2022. URL: https://openreview.net/forum?id=k5sdO59Pw_3.

[Sta23] Statista.com. *Online Food Delivery - United States*. https://www.statista.com/outlook/dmo/online-food-delivery/united-states. [Online; accessed 01-February-2024]. 2023.