Facultat de Matemàtiques i Estadística
Universitat Politècnica de Catalunya

Master's Thesis

SOLUTION OF A MULTI-DEPOT, HETEROGENEOUS FLEET VEHICLE ROUTING PROBLEM USING COLUMN GENERATION

Alejandro Arenas Vasco

Director: Elena Fernández Aréizaga
Jessica Rodríguez Pereira

Department of Statistics and Operations Research

Thanks to Esteban, Nanita, los cuchos and Tothom for the support.
And to Elena and Jess for the knowledge.

Abstract

For the enterprises that have to serve different clients with a fleet, the vehicle routing decisions are tactically critical. The main reason: the efficiency of the whole company is at stake given the cost of the resources involved (vehicles) and the operational costs of the whole process (driver's salary, fuel).

This thesis addresses three different types of problems related to the vehicle routing decisions. The first problem is the simplest one where all the vehicles are of the same type (a homogeneous fleet) and all the routes start and end at the same depot. These characteristics make the routes easier to design. Complexity is added when, instead of one depot, multiple depots can be used. This gives the enterprise more flexibility to serve the clients but increases exponentially the number of possible routes a vehicle can take. Lastly, the possibility of using different types of vehicles (a heterogeneous fleet) makes the routing problem even more complex.

Due to the difficulty of the route scheduling, standard methods to solve optimization problems fail to deliver good results when the number of clients, depots and types of vehicles increase. For small instances (seven clients), solution methods based on standard mixed-integer programming formulations perform optimally but when the number of clients increases such methods are no longer effective. Because of this a different approach has to be taken to obtain good results. Column generation methods are proposed in this thesis.

Column generation methods start with a feasible solution to solve a Restricted Master Problem. In the case of vehicle routing problems this is typically developed through a Dantzig-Wolfe decomposition of the routing decision variables. Then the dual variables of the initial problem are used to find new routes that can improve the former problem. These routes are added to the initial ones considered in the Restricted Master Problem and the process is repeated. The algorithm finishes when there are not more routes that can be added to improve the current problem. The key factor in this method is that only a subset of variables is used instead of all of them.

The objective of this Master thesis is to describe the three vehicle routing problems, the classical formulation of each one and the column generation formulation. Then, a practical application based upon a Colombian company is used to generate different data sets to: compare the performance of the column generation method with the standard one, to compare the behaviour of the method within different kind of data sets, and to evaluate how the method performs in large scale problems.

Keywords: Vehicle Routing Problem, Multi Depot Vehicle Routing Problem, Multi Depot Multi Vehicle Routing Problem, Heterogeneous Fleet, Column Generation, Dantzig-Wolfe.

Resumen

En las empresas que deben atender diferentes clientes usando una flota de vehículos, las decisiones relacionadas con el diseño de las rutas son tácticamente críticas. La razón estriba en que la eficiencia de toda la compañía está en juego dado los altos costos de los recursos involucrados (vehículos) y los diferentes costos operacionales del proceso (salario de conductores, gasolina).

Esta tesis se enfoca en tres tipos de problemas diferentes relacionados con el diseño de las rutas de vehículos. El primero (el más sencillo) es aquél en el cual todos los vehículos son del mismo tipo (flota homogénea) y todas las rutas salen y regresan al mismo depósito. Estas características permiten que las rutas sean más sencillas de programar. Al agregar la posibilidad de usar distintos depósitos, se da más flexibilidad a la empresa pero el problema también se torna más complejo ya que la cantidad de posibles rutas aumenta exponencialmente. Por último, se agrega la posibilidad de usar diferentes tipos de vehículos (flota heterogénea) haciendo el problema aún más complejo.

Debido a la complejidad del problema, los métodos estándar para solucionar problemas de optimización son incapaces de proporcionar buenos resultados a medida que aumentan el número de clientes, de depósitos y el tipo de vehículos. Para pequeñas instancias (siete clientes), métodos de solución basados en formulaciones de programación matemática entera mixta estándar operan de manera adecuada, sin embargo, al aumentar el número de clientes a 12, se pierde este adecuado funcionamiento. Debido a lo anterior, es necesario utilizar otro método para obtener buenos resultados. El método que se propone en esta tesis para este propósito está basado en generación de columnas.

Los métodos de generación de columnas inician con una solución factible a partir de la cual se soluciona un Problema Maestro Restringido. En el caso de problemas de rutas de vehículos, dicho problema se deriva típicamente descomponiendo las variables de ruteo usando Dantzig-Wolfe. Posteriormente se usan las variables duales del problema anteriormente descrito para buscar nuevas rutas que permitan mejorar el valor del mismo. Una vez encontradas dichas rutas, estas se incorporan al Problema Maestro Restringido y el proceso se repite. El algoritmo termina cuando no existen más rutas que puedan mejorar el Problema Maestro Restringido. El factor clave en este método es que sólo requiere un subgrupo de variables en lugar de todas ellas.

El objetivo de esta tesis de Maestría es estudiar los tres problemas de ruteo de vehículos enunciados previamente, comparando sus respectivas formulaciones clásicas con formulaciones alternativas adecuadas para ser tratadas mediante generación de columnas. Posteriormente, se utiliza el caso de una empresa colombiana para, en base a sus problemas reales, crear diferentes sets de datos con el fin de: comparar el desempeño del método de generación de columnas con el método estándar, comparar el comportamiento del método de generación de columnas entre sí con diferentes sets de datos y, por último, evaluar como se desempeña el método en problemas a gran escala.

Palabras clave: Problemas de Ruteo de Vehículos, Problemas de Ruteo de Vehículos con Múltiples Depósitos, Problemas de Ruteo de Vehículos con Múltiples Depósitos y Múltiples tipos de Vechículos, Flota Heterogénea, Generación de Columnas, Dantzig-Wolfe.

Notation

A: Set of arcs
C: Set of customers
D: Set of depots
G: Complete and directed graph
K: Set of vehicles
MDMVRP: Multi Depot Multi Vehicle Routing Problem
MDVRP: Multi Depot Vehicle Routing Problem
MP: Master Problem
N: Set of nodes
OP: Original Problem
RMP: Restricted Master Problem
SP: Subproblem
VRP: Vehicle Routing Problem
VV: Vending Venta S.A.

Index

Introduction

In the Colombian industry environment, it is quite common to solve difficult problems using a trial and error method. More usual than not, the approach to determine the solution to be implemented is only empirical knowledge. This is especially true for enterprises that have to program a fleet of vehicles to serve different clients. The main reason is that it has always been done this way and, so far, the results have been adequate.

What Colombian enterprises are usually not taking into account is the fact that expanding a routing network adds a lot of different routing possibilities. Even if the person in charge of the routing schedule has a lot of experience, the number of possible routes to program are exponential and impossible to imagine by a human being. This makes the possibility of solving the routing problem using a systematical approach a necessity. Thus, the motivation of this Master thesis is developing a rigorous framework for addressing some real applications of Vehicle Routing Problems.

These Vehicle Routing Problems can be as general as a simple network with clients, a unique depot and a standard vehicle to serve the demand. Or can be as specific as a network with clients with different demands and a timeframe to be served, different depots to serve the demand, different types of vehicles and even more details. The limits of this Master thesis are an intermediate point in which the clients have a specific demand, multiple depots can be used and different types of vehicles are available. The reason is that the enterprise which was taken as an example for the project works under these premises.

After reading this thesis, the reader will have arguments to pursue a systemic approach to Vehicle Routing Problems instead of a pure trial and error method.

In the Chapter 1 of this text, all the framework of the thesis is explained. Initially the classical Vehicle Routing Problem is formulated using a classical approach and then a column generation approach. Then, in Chapter 2, the possibility of using multiple depots first and multiple type of vehicles second are added to the problem. The classical formulation and the column generation formulation are described for both cases. In Chapter 3, the results of a practical problem are presented and analyzed. In the last chapter the conclusions are presented.

Chapter 1

Framework

This chapter describes the methodological framework for this Master's thesis, which are the VRPs and column generation methods.

1. The Vehicle Routing Problem

In the Vehicle Routing Problem (VRP), a set of clients must be served from a depot using a fleet of vehicles (of the same type initially). Each client has a given demand and each vehicle a limit of units it can carry. There is the possibility of using more than one depot and more than one type of vehicle. These options will be explored further on.

1.1 Input and Sets

The simple VRP is designed upon a connected and directed Graph *G = (N,A),* where *N* is the set of nodes and *A* is the set of arcs. The set of clients *C* and a depot *0* ($N = C \cup 0$) is also given. The cost of using the arc *(i,j)* is represented by C_{ij} (because the fleet is homogeneous, the cost is the same for all the vehicles). The index set of available vehicles to serve the demand is *K* and the demand of every client is d_i. Lastly, as this is a homogeneous fleet, the capacity, *q,* of all the vehicles is equal.

1.2 Variables

In order to formulate the VRP with a mixed-integer mathematical programming formulation, two families of variables are usually considered:

1. $X_{ij}^{k} \in \{0,1\}$: Binary variable takes the value 1 if the arc *(i,j)* is used by the k^{th} vehicle and zero otherwise. The number of variables of this family is $|K||N^2|$.
2. $U_i \geq 0$: Is an auxiliary variable which represents the position in which node i is visited on its route. The objective of this variable is to ensure that the routes of each vehicle are well-defined and avoid cycling. The number of variables of this family is $|N|$.

1.3 Formulation of the VRP

In the VRP, the objective is to minimize the total cost of the system having served every client in the network:

$$VRP \coloneqq \min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij} X_{ij}^{k} \quad (1.1)$$

$subject\ to$

$$\sum_{k \in K} \sum_{j \in N} X_{ij}^{k} \geq 1 \qquad \forall i \in C \quad (1.2)$$

$$\sum_{i \in C} d_i \sum_{j \in N} X_{ij}^{k} \leq q \qquad \forall k \in K \quad (1.3)$$

$$\sum_{j \in N} X_{0j}^{k} = 1 \qquad \forall k \in K \quad (1.4)$$

$$\sum_{i \in N} X_{i0}^{k} = 1 \qquad \forall k \in K \quad (1.5)$$

$$\sum_{i \in N} X_{ih}^{k} - \sum_{j \in N} X_{hj}^{k} = 0 \qquad \forall h \in C, \forall k \in K \quad (1.6)$$

$$U_0 = 1 \quad (1.7)$$

$$U_j \geq (U_i + 1) - M\left(1 - \sum_{k \in K} X_{ij}^{k}\right) \qquad \forall i \in N, \forall j \in C \quad (1.8)$$

$$X_{ij}^{k} \in \{0,1\} \qquad \forall i,j \in N, \forall k \in K \quad (1.9)$$

$$2 \leq U_i \leq |N| \qquad \forall i \in C \quad (1.10)$$

Constraints (1.2) ensure that every client is visited at least once. Inequalities (1.3) limit the amount of clients served by a vehicle according to their capacity. Constraints (1.4) and (1.5) guarantee that the vehicles depart and finish the route at the depot 0. Equalities (1.6) assure that the flow continues through the network. Equation (1.7) is used to secure that the first node to be visited according to auxiliary variables U is the depot. Inequalities (1.8) determine that the value of U_j must be higher than U_i, when i is the previously visited node. These two last sets of constraints avoid the possibility of having subcircuits in the routes. The last two equations define the domain of the variables.

2. Column generation

The general idea behind column generation (CG) is to define a mathematical programing formulation for a problem that considers an arbitrarily large number of decision variables. Normally, to solve optimally the considered problem, a subset of the variables is enough. Hence, a Restricted Master Problem (RMP) is solved with only a subset of variables and an auxiliary pricing subproblem identifies additional variables that may help improve the current solution if they exist (Lübbecke 2010). The criteria to pick the new variables to add in the model is dictated by the pricing subproblem (SP). The objective of the SP is to find a new variable with reduced negative cost to improve the result of the RMP.

2.1 The Master Problem

Consider an optimization problem (OP) of the form:

$$OP \coloneqq \min c^T \boldsymbol{x} \quad (1.11)$$

subject to

$$Ax \geq b \tag{1.12}$$
$$Dx \geq d \tag{1.13}$$
$$x \geq 0 \tag{1.14}$$

where we are implicitly assuming that $D\boldsymbol{x} \geq d$ is a set of difficult constraint whereas the other two sets are easy to satisfy. Let P be the Polyhedra associated with the x vectors that satisfy the difficult constraints together with the non-negativity conditions, that is $P = \{x \in \mathbb{R}_+^n \; | D\boldsymbol{x} \geq d\}$. Then x can be decomposed as (Schrijver 1986):

$$x = \sum_{q \in Q} p_q y_q + \sum_{r \in R} p_r y_r, \quad \sum_{q \in Q} y_q = 1, \qquad y \in \mathbb{R}_+^{|Q|+|R|} \tag{1.15}$$

where p_q are extreme points and p_r are extreme rays of P and Q and R are finite. That is, any feasible solution to OP can be expressed as a convex combination of extreme points and extreme rays of P, which additionally satisfies the set of *easy* constraints. Thus OP can be rewritten as the Master Problem (MP):

$$MP := \min \sum_{q \in Q} c_q y_q + \sum_{r \in R} c_r y_r \tag{1.16}$$

subject to

$$\sum_{q \in Q} a_q y_q + \sum_{r \in R} a_r y_r \geq b \tag{1.17}$$
$$\sum_{q \in Q} y_q = 1 \tag{1.18}$$
$$\boldsymbol{y} \geq 0 \tag{1.19}$$

There are some important features of MP that are important to be considered:

- c_q is the cost associated with the extreme point $q \in Q$.
- c_r is the cost associated with the extreme ray $r \in R$.
- MP and OP have the same objective function value but the Polyhedra of each of them is different (Nazareth 1987).
- The components of y identify the extreme points and extreme rays that determine the different feasible solutions, as well as their weights in the convex combination.

2.2 The Restricted Master Problem

The difficulty with MP is that it involves a large number of variables, since $|Q| + |R|$ is very large. Even though the number of variables of MP is usually smaller than in OP (Lübbecke et. al. 2005), for big instances it is too difficult to solve with standard methods. So a RMP is suggested. In RMP the formulation is the same as in MP but, instead of using all the possible extreme points and extreme rays, a subset of them is used ($|Q'| + |R'|$):

$$RMP \coloneqq \min \sum_{q \in Q'} c_q y_q + \sum_{r \in R'} c_r y_r \tag{1.20}$$

subject to

$$\sum_{q \in Q'} a_q y_q + \sum_{r \in R'} a_r y_r \geq b \tag{1.21}$$

$$\sum_{q \in Q'} y_q = 1 \tag{1.22}$$

$$\boldsymbol{y} \geq 0 \tag{1.23}$$

Because in RMP only a subset of the possibilities is being used, the optimal solution of RMP is not necessarily the optimal solution of MP. It does not even produce a valid lower bound for OP. In order to improve the solution, a pricing subproblem (SP) is needed.

2.3 The Pricing Subproblem

Let us consider an optimal solution for RMP. The dual multipliers for the two constraints will be called π_1 of (1.21) and π_2 of (1.22). A negative reduced cost will mean that a new set of variables will improve the value of the RMP. But, these new set of variables, must be feasible in the difficult constrain $D\boldsymbol{x} \geq d$ in OP. So the next SP must be solved:

$$SP \coloneqq \min(c^T - \pi_1^T A)x - \pi_2 \tag{1.24}$$

subject to

$$D\boldsymbol{x} \geq d \tag{1.25}$$

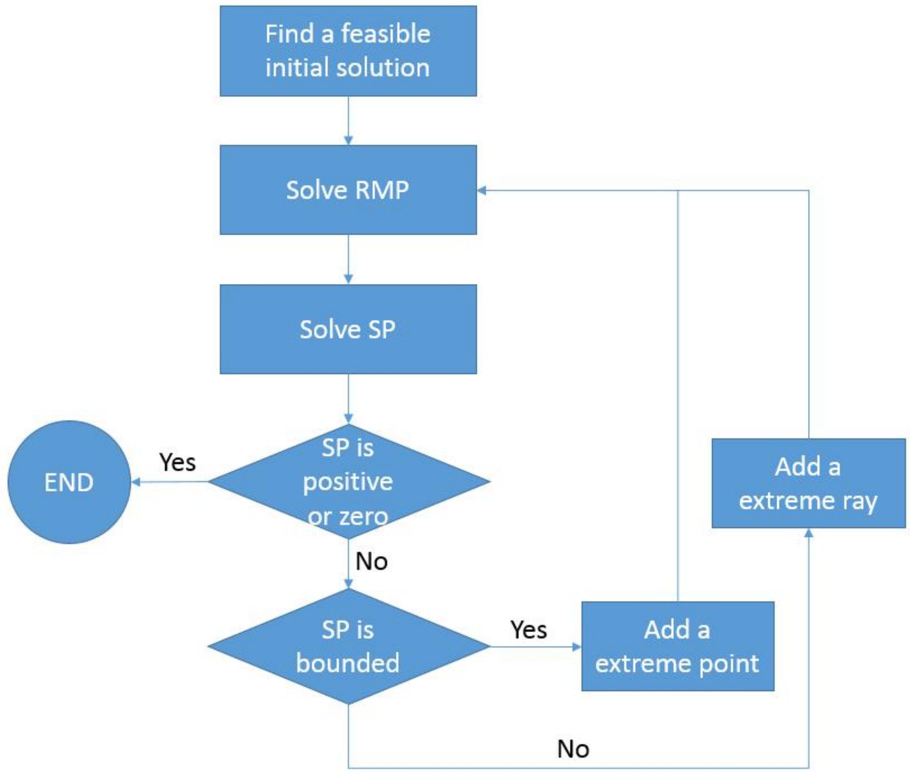
$$\boldsymbol{x} \geq 0 \tag{1.26}$$

The SP can have three different outcomes:

- The optimal value is non-negative: There is no reduced cost that may improve RMP, so the current subsets of Q and R define the optimal solution.
- The optimal value is negative but finite: The new set of extreme points reduces the value of RMP so it must be added to RMP.
- SP is unbounded below: The solution of SP is an extreme ray and it must be added to RMP.

2.4 Method Overview

Column generation is an iterative method which consists in adding feasible solutions to RMP until its optimal value cannot be improved:



Picture 1. Column generation scheme

This method allows the user to find an optimal solution without having to process the entire spectrum of feasible solutions. This property is very useful for problems with a lot of variables.

2.5 Column generation formulation for the VRP

For the column generation approach to be applied in the VRP, only the constraint that guarantees the visit of every client (1.2) will be left in MP. The others will be taken into account in the SP.

To solve the linear programming relaxation of the VRP using Column Generation the X variables must be reformulated. Using (1.15), they must be expressed as a combination of extreme points and extreme rays. Nevertheless, in this formulation X is only the combination of different paths (which are feasible) so there is no need to consider extreme rays (Kallehauge et. al. 2005). Considering P^k as the set of possible paths for vehicle *k*:

$$X_{ij}^k = \sum_{p \in P^k} X_{ijp}^k y_p^k \qquad \forall k \in K, \forall (i,j) \in A \quad (1.27)$$

$$\sum_{p \in P^k} y_p^k = 1 \qquad \forall k \in K \quad (1.28)$$

$$y_p^k \geq 0 \qquad \forall k \in K, \forall p \in P^k \quad (1.29)$$

With this transformation of variable X, the cost of using a particular path in a particular vehicle can be expressed as:

$$C_p^k = \sum_{i \in N} \sum_{j \in N} C_{ij}^k X_{ijp}^k \qquad \forall k \in K, \forall \mathrm{p} \in \mathrm{P^k} \qquad (1.30)$$

Using the same logic, a new parameter to calculate the number of times a customer is visited in path *p* in vehicle *k* can be written as:

$$a_{ip}^k = \sum_{j \in N} X_{ijp}^k \qquad \forall k \in K, \forall i \in N, \forall \mathrm{p} \in \mathrm{P^k} (1.31)$$

Equation (1.30) can be replaced in the objective function of the VRP. In the same line of thought, (1.31) can be used to reformulate constraint (1.2) thus obtaining the MP for the VRP:

$$MP \coloneqq min \sum_{k \in K} \sum_{p \in P^k} C_p^k y_p^k \qquad (1.32)$$

$subject\ to$:

$$\sum_{p \in P^k} \sum_{k \in K} a_{ip}^k y_p^k \geq 1 \qquad \forall i \in C \qquad (1.33)$$

$$\sum_{p \in P^k} y_p^k = 1 \qquad \forall k \in K \qquad (1.34)$$

$$y_p^k \geq 0 \qquad \forall p \in P^k, \forall k \in K \qquad (1.35)$$

Before developing the reasoning for the SP, there is something important to remark about the MP. The meaning of variable y_p^k (which is not binary) is the fraction of the path *p* used in vehicle *k*. Because this is a simple VRP with only one type of vehicle $P^1 = P^2 = \cdots = P^K = P$ thus index *k* is utterly unnecessary. Nevertheless, to eliminate index *k*, constraint $\sum_{p \in P^k} y_p^k = 1$ now changes to $\sum_{p \in P} y_p \leq |K|$ which guarantees that the number of paths used is less or equal than the total number of vehicles available.

Taking all of this into consideration and using only a subset P' of all the paths available, the RMP can be stated:

$$RMP_{VRP} \coloneqq min \sum_{p \in P'} C_p y_p \qquad (1.36)$$

$subject\ to$:

$$\sum_{p \in P'} a_{ip} y_p \geq 1 \qquad \forall i \in C \qquad (1.37)$$

$$\sum_{p \in P'} y_p \leq |K| \qquad (1.38)$$

$$y_p \geq 0 \qquad \forall p \in P' \qquad (1.39)$$

For the SP, let's consider π_{1i} and π_2 as the dual variables for the two constraints in formulation (1.36-1.39). The first dual variable can directly affect the cost matrix as follows: $\widehat{C_{ij}} = C_{ij} - \pi_{1i}$ meanwhile π_2 affects directly the objective function. The SP formulation is:

$$SP_{VRP} \coloneqq min \sum_{(i,j)\in N} \widehat{C_{ij}} X_{ij} + \pi_2 \tag{1.40}$$

subject to:

$$\sum_{i\in C} d_i \sum_{j\in N} X_{ij} \leq q \tag{1.41}$$

$$\sum_{j\in N} X_{0j} = 1 \tag{1.42}$$

$$\sum_{i\in N} X_{i0} = 1 \tag{1.43}$$

$$\sum_{i\in N} X_{ih} - \sum_{j\in N} X_{hi} = 0 \qquad \forall h \in C \tag{1.44}$$

$$U_0 = 1 \tag{1.45}$$

$$U_j \geq (U_i + 1) - M(1 - X_{ij}) \qquad \forall i \in N, \forall j \in C \tag{1.46}$$

$$X_{ij} \in \{0,1\} \qquad \forall i,j \in N \tag{1.47}$$

$$2 \leq U_i \leq |N| \qquad \forall i \in C \tag{1.48}$$

The SP basically has the purpose of finding a new feasible path which will improve the value of RMP. All the constraints are the same as in the formulation in the VRP which are not included in RMP to assure the feasibility of the path. Because there is only one type of vehicle and one depot, there is only one SP to be solved. The algorithm stops when the objective function of the SP is positive (then RMP can't be improved) or after a determined consecutive number of iteration without improving RMP.

Chapter 2

Multi Depot Multi Vehicle Routing Problem

This chapter describes the Multi Depot Multi Vehicle Routing Problem (MDMVRP) with a heterogeneous fleet. For that, previously, as an intermediate step, the Multi Depot Vehicle Routing Problem (MDVRP) with homogeneous fleet must be studied as well. The column generation approach is attached to both problems.

1. Multi Depot Vehicle Routing Problem

Adding the possibility of attending the demand from multiple depots increases considerable complexity to the model. For this particular case, we assume the routes must end in the same depot that they started. As a consequence, and in relation with the VRP, a new family of variables to assign every vehicle to the origin depot must be created. It is important also to consider that the depots have finite capacity to attend a certain number of vehicles.

1.1 Input and Sets

The MDVRP is designed upon a connected and directed Graph *G = (N,A),* where *N* is the set of nodes and *A* is the set of arcs. The set of clients *C* and of depots *D* is given ($N = C \cup D$). The cost of using the arc *(i,j)* is represented with C_{ij}. The set of available vehicles to attend the demand is *K*, as this is a homogeneous fleet, the capacity, *q,* of all the vehicles is equal. The demand to satisfy of every client is d_i. Lastly, as the depots have a finite capacity to attend the vehicles, the maximum capacity of depot *i* is added, $maxdepo_i$.

1.2 Variables

The MDVRP keeps the two family of variables introduced in the VRP and incorporates a new one:

1. $X_{ij}^k \in \{0,1\}$: Binary variable takes the value 1 if the arc (*i,j*) is used by the k^{th} vehicle and zero otherwise. The number of variables of this family is $|K||N^2|$.
2. $U_i \geq 0$: Is an auxiliary variable which represents the position in which node i is visited on its route. The objective of this variable is to ensure that the routes of each vehicle are well-defined and avoid cycling. The number of variables of this family is $|C|$.
3. $W_d^k \in \{0,1\}$: Binary variable takes the value 1 if vehicle *k* is served from depot *d* and zero otherwise. The number of variables of this family is $|K||D|$.

1.3 The formulation of the MDVRP

The formulation of the MDVRP follows the same ideas as the one explained before for the VRP, and in general for the routing problems. It is necessary to ensure well-defined routes. The difference is that now constraints (1.4) and (1.5) which ensure that all vehicles depart and finish the route at the depot 0, are subject to variable W in constraints (2.4) and (2.5). Thus, if a vehicle is not associated to a depot, it can't depart from that depot or return to that depot. Also, a new family of inequalities (2.7) is needed to guarantee that the capacity of the depot is not violated.

$$\text{MDVRP} := \min \sum_{k\in K}\sum_{i\in N}\sum_{j\in N} C_{ij}X_{ij}^{k} \tag{2.1}$$

$subject\ to$:

$$\sum_{k\in K}\sum_{j\in N} X_{ij}^{k} = 1 \qquad \forall i \in C \tag{2.2}$$

$$\sum_{i\in C} d_i \sum_{j\in N} X_{ij}^{k} \leq q \qquad \forall k \in K \tag{2.3}$$

$$\sum_{j\in N} X_{hj}^{k} = W_h^k \qquad \forall h \in D, \forall k \in K \tag{2.4}$$

$$\sum_{i\in N} X_{ih}^{k} = W_h^k \qquad \forall h \in D, \forall k \in K \tag{2.5}$$

$$\sum_{i\in N} X_{ih}^{k} - \sum_{j\in N} X_{hj}^{k} = 0 \qquad \forall h \in C, \forall k \in K \tag{2.6}$$

$$\sum_{k\in K} W_i^k \leq maxdepo_i \qquad \forall i \in D \tag{2.7}$$

$$U_i = 1 \quad \forall i \in D \tag{2.8}$$

$$U_j \geq (U_i + 1) - M\left(1 - \sum_{k\in V} X_{ij}^{k}\right) \qquad \forall i \in N, \forall j \in C \tag{2.9}$$

$$X_{ij}^{k} \in \{0,1\} \qquad \forall i,j \in N, \forall k \in K \tag{2.10}$$

$$2 \leq U_i \leq |N| \qquad \forall i \in C \tag{2.11}$$

1.4 Column generation Formulation for the MDVRP

The process to derive RMP in the MDVRP model follows the same idea to the process to deduce RMP_{VRP}. The only difference is that the new variable that limits the amount of vehicles that depart the depots is now added. For that, parameter X_{ijp} is used to know if path p uses any arc that starts in the depots. Note that X_{ijp} is a parameter because is information about the paths that conform P'.

$$RMP_{MDVRP} := min \sum_{p\in P'} C_p y_p \tag{2.12}$$

subject to:

$$\sum_{p \in P'} a_{ip} y_p \geq 1 \qquad \forall i \in C \qquad (2.13)$$

$$\sum_{p \in P'} y_p \leq |K| \qquad (2.14)$$

$$\sum_{i \in N} \sum_{p \in P'} X_{ijp} y_p \leq maxdepo_i \qquad \forall i \in D \qquad (2.15)$$

$$y_p \geq 0 \qquad \forall p \in P' \qquad (2.16)$$

In the formulation of this RMP_{MDVRP}, a new dual variable appears: π_{d3}. This dual variable is related to the capacity of each of the depot (2.15). As a consequence of this, for every depot a different SP must be solved.

For the SP formulation, let's consider ב as the depot for which the SP is being solved. Also, N' will be a set with all the clients and the depot ב. ($N' \subset N$). As before, $\widehat{C_{ij}} = C_{ij} - \pi_{1i}$.

$$SP_{MDVRP} \coloneqq min \sum_{(i,j) \in N'} \widehat{C_{ij}} X_{ij} + \pi_2 + \pi_{ב3} \qquad (2.17)$$

subject to:

$$\sum_{i \in C} d_i \sum_{j \in N'} X_{ij} \leq q \qquad (2.18)$$

$$\sum_{j \in N'} X_{בj} = 1 \qquad (2.19)$$

$$\sum_{i \in N'} X_{iב} = 1 \qquad (2.20)$$

$$\sum_{i \in N'} X_{ih} - \sum_{j \in N'} X_{hi} = 0 \qquad \forall h \in C \qquad (2.21)$$

$$U_j \geq (U_i + 1) - M(1 - X_{ij}) \qquad \forall i \in N', \forall j \in C \qquad (2.22)$$

$$U_ב = 1 \qquad (2.23)$$

$$X_{ij} \in \{0,1\} \qquad \forall i,j \in N' \qquad (2.24)$$

$$2 \leq U_i \leq |N'| \qquad \forall i \in C \qquad (2.25)$$

Now, the purpose of the SP, is to add to P' in the RMP_{MDVRP} as many new paths as depots which can improve the optimal value of it. The criteria to stop the algorithm is when the optimal value for the SP_{MDVRP} in all the depots are positive or after a determined consecutive number of iterations without improving RMP_{MDVRP}.

2. Multiple Depot Multiple Vehicle Routing Problem

Finally, the possibility of using a heterogeneous fleet with different types of vehicles which may have different or costs is added. This addition affects slightly the normal formulation and the column generation formulation. However, it adds complexity to the algorithm to solve the problem.

2.1 Input and Sets

The MDMVRP is designed upon a connected and directed Graph *G = (N,A),* where *N* is the set of nodes and *A* is the set of arcs. The set of clients *C* and of depots *D* are needed ($N = C \cup D$). The cost of using the arc *(i,j)* is represented with C_{ij}^{k}. Note that now the costs are different according to the type of vehicle so, index k is needed. The set of available vehicles to attend the demand is *K,* as this is not a heterogeneous fleet, the capacity of all the vehicles is q_k. The demand of every client is d_i. Lastly, as the depots have a finite capacity to attend the vehicles, the maximum capacity of depot *i* is added, $maxdepo_i$.

Two new inputs are needed only for the column generation approach. The first of them is the set *T* which defines the types of vehicles available. The second, α_t is the amount of available vehicles of each type.

2.2 Variables and formulation of the MDMVRP

The formulation of the MDVRP can be easily extended to model the MDMVRP using the exactly the three same sets of variables. Only in (2.28) the index of the capacity is added. Thus, the formulation is:

$$\text{MDMVRP} \coloneqq \min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij}^{k} X_{ij}^{k} \qquad (2.26)$$

$subject\ to$:

$$\sum_{k \in K} \sum_{j \in N} X_{ij}^{k} = 1 \qquad \forall i \in C \qquad (2.27)$$

$$\sum_{i \in C} d_i \sum_{j \in N} X_{ij}^{k} \leq q_k \qquad \forall k \in K \qquad (2.28)$$

$$\sum_{j \in N} X_{hj}^{k} = W_h^k \qquad \forall h \in D, \forall k \in K \qquad (2.29)$$

$$\sum_{i \in N} X_{ih}^{k} = W_h^k \qquad \forall h \in D, \forall k \in K \qquad (2.30)$$

$$\sum_{i \in N} X_{ih}^{k} - \sum_{j \in N} X_{hj}^{k} = 0 \qquad \forall h \in C, \forall k \in K \qquad (2.31)$$

$$\sum_{j \in N} W_i^k \leq maxdepo_i \qquad \forall i \in D \qquad (2.32)$$

$$U_i = 1 \qquad \forall i \in D \qquad (2.33)$$

$$U_j \geq (U_i + 1) - M\left(1 - \sum_{k \in V} X_{ij}^{k}\right) \qquad \forall i \in N, \forall j \in C \qquad (2.34)$$

$$X_{ij}^{k} \in \{0,1\} \qquad \forall i,j \in N, \forall k \in K \qquad (2.35)$$

$$2 \leq U_i \leq |N| \qquad \forall i \in C \qquad (2.36)$$

2.3 Formulation of the MDMVRP to use in column generation

The formulation of RMP can be easily extended to the RMP_{MDMVRP}. Nevertheless, as presented in (1.27-1.29), it is now necessary to add indexes according to the type of vehicle (*t*) to all the variables and parameters:

$$RMP_{MDMVRP} \coloneqq min \sum_{t \in T} \sum_{p \in P'} C_p^t y_p^t \tag{2.37}$$

$subject\ to$:

$$\sum_{t \in T} \sum_{p \in P'} a_{ip}^t y_p^t \geq 1 \qquad \forall i \in C \tag{2.38}$$

$$\sum_{p \in P'^k} y_p^t \leq \alpha_t \qquad \forall t \in T \tag{2.39}$$

$$\sum_{i \in N} \sum_{p \in P'} \sum_{t \in T} X_{ijp}^t y_p^t \leq maxdepo_i \qquad \forall i \in D \tag{2.40}$$

$$y_p \geq 0 \qquad \forall p \in P' \tag{2.41}$$

The second dual variable of RMP_{MDMVRP} is now indexed (π_{t2}) as it depends on the type of vehicle being used. As a consequence, now a SP must be solved for every combination of type of vehicle and depot ($|T||D|$ number of SP to be solved).

For the SP formulation, let's consider ב as the depot for which the SP is being solved and א the type of vehicle for which the SP is being solved. Also, N' will be a set with all the clients and the depot ב. ($N' \subset N$). Now, there is a difference in the calculation of the cost $\widehat{C_{ij}}$. Because each type of vehicle has different cost, the dual cost must use only the type of vehicle which is being analyzed in the SP: $\widehat{C_{ij}} = C_{ij}^{\aleph} - \pi_{1i}$.

$$SP_{MDMVRP} \coloneqq min \sum_{(i,j) \in N'} \widehat{C_{ij}} X_{ij} + \pi_{\aleph 2} + \pi_{\beth 3} \tag{2.42}$$

$subject\ to$:

$$\sum_{i \in C} d_i \sum_{j \in N'} X_{ij} \leq q_{\aleph} \tag{2.43}$$

$$\sum_{j \in N'} X_{\beth j} = 1 \tag{2.44}$$

$$\sum_{i \in N'} X_{i \beth} = 1 \tag{2.45}$$

$$\sum_{i \in N'} X_{ih} - \sum_{j \in N'} X_{hi} = 0 \qquad \forall h \in C \tag{2.46}$$

$$U_{\beth} = 1 \tag{2.47}$$

$$U_j \geq (U_i + 1) - M(1 - X_{ij}) \qquad \forall i \in N', \forall j \in C \tag{2.48}$$

$$X_{ij} \in \{0,1\} \qquad \forall i,j \in N' \tag{2.49}$$

$$2 \leq U_i \leq |N'| \qquad \forall i \in C \tag{2.50}$$

Chapter 3

Results and Analysis

The first section of this chapter describes a practical application for the framework explained in Chapter 1. This is important for the rest of the chapter because the datasets used in the other sections are based upon this information. Then, a summary of the column generation method used in this thesis will be explained. Finally, some comparisons are made. The first between the traditional method and the column generation method. The second between the results of the column generation method when difficulty increases. And last the scope is put in a large scale problem.

1. Practical application: Vending Venta S.A.

The real application is based in one of the biggest enterprises in the vending machines cluster in Medellín, Colombia. Due to privacy policy the name is kept anonymous, and we will refer to it as Vending Venta (VV). The business idea of VV is to manage three different types of vending machines in different spots along the city. This vending machines can be of hot beverages (coffee, cocoa), cold beverages (water, tea) and snacks (chocolate bars, popcorn).

Picture 2. Vending Machine example

1.1 Operation of Vending Venta

Everything starts with a commercial agent of VV deciding where to allocate or relocate a vending machine in Medellín. Then, an analyst decides in a daily basis the quantities and the products to be packed in every machine. Logistics processes the orders in the afternoon in the depot so that they are ready next day in the morning. The personnel that must re-fill the machines every day are called routers. The routers can either work in pair or alone depending of the vehicle they are assigned to. In that sense, there are two possible vehicles which are described below.

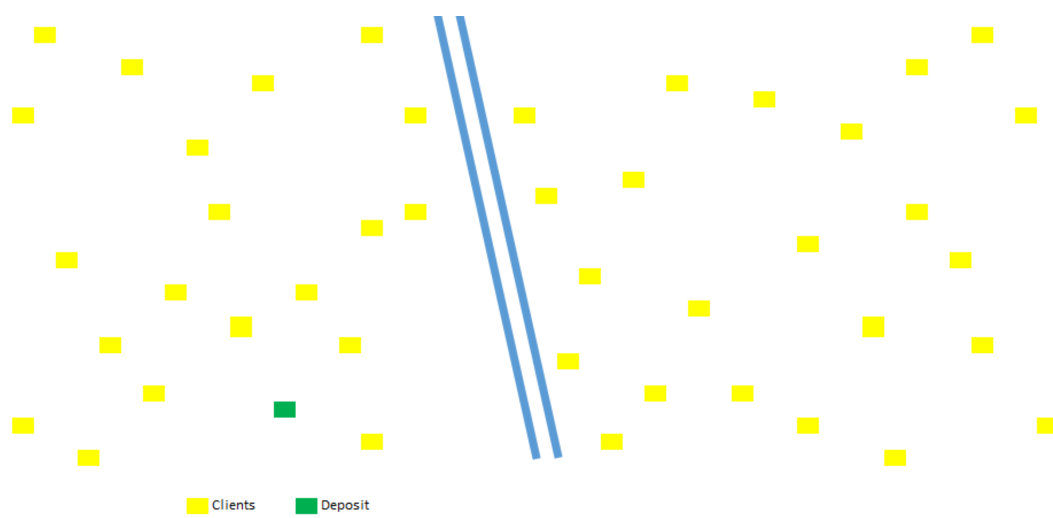
Picture 3. Vehicles available in VV

- Motocarrier (left in the picture): The motocarriers are motorbikes adapted to transport goods. They are cheaper to use than the vans but have less capacity (given in quantity of vending machines that can be attended). The motocarrier can transport one router and the merchandise for 12 machines. Because the motocarriers have a smaller motor than the vans and can move easily in urban areas, the cost of transportation is smaller than the cost of transportation for the vans.
- Van (right in the picture): The vans have more capacity than the motocarriers (up to 20 machines) and need two routers to operate. As it needs more personnel and are less fuel-efficient thus the cost of using is higher than the cost of using a motocarrier.

The demand of a vending machine is stochastic (when a router arrives to re-fill it, it isn't necessarily empty). As a consequence, around 15% of the products packed to serve the machines have to be returned to the depot so logistics can repack them for the next day. The immediate consequence of this process is that the routers have to finish the route in the depot.

1.2 Volume data

VV has in the city of Medellin more than 250 machines in 50 establishments (as a rule of thumb, there is always more than one machine per place for profitability reasons). The company has at the moment one depot but is looking forward to build at least another one. One of the big problems for the company is that the majority of the clients are located in the other side of the Medellin River. VV has 18 vehicles available to serve the demand ten of which are vans and the rest motocarriers.



Picture 4. Map of the deposit and clients of VV

1.3 Geographical context

The city of Medellín is inhabited by approximately 4 million people that share 382 km^2 of land. The city is divided by the Medellín River which has a big impact in the city's mobility. Due to the city having only seven bridges to cross the river, normally this becomes a bottleneck in the routing scheduling.

Picture 5. The Medellin River

Another important fact in the routing scheduling is the actual road network infrastructure used on the city. It can't withstand all the vehicles that, in an uncontrolled way, are added to the road network every day (Posada et. al. 2011). As a consequence, transiting Medellín is usually a slow and expensive process.

2. Relation of VV with the VRP

The operation in VV in the present resembles the VRP with multiple types of vehicles. But in the near future, with the new depots that the company is going to open, it will resemble a MDMVRP problem.

2.1 Actual process

Every day in VV, the route to be chosen for the vending machine re-fill is picked by the routers according to their experience. As a result, there is a lack of standardization in the process and every router takes different paths with different results.

Besides, there is another structural problem which affects the operation of VV. Every day, each router is given a fixed amount of vending machines to re-fill. The process of selecting which machines should be filled by every router is based upon geographical proximity. Normally this is done without taking into account the cost of using arcs due to the lack of resources to do so. As a result, there are routes that have to deal with two or more crossings of the Medellín River thus affecting the efficiency of the process.

With all the information available, the process of routing can be efficiently scheduled for all the company using a VRP with a multi vehicles approach.

2.2 Future process

In the near future, VV will have at least another depot in the other side of the river. There is a possibility of building a third one but this will be for the long term. With a second depot, the scheduling of the re-fill of the vending machines can be efficiently solved using the MDMVRP exposed in Chapter 1.

Another idea of VV is to add another type of vehicle besides the motocarriers and vans. This a truck for heavy loads which will be able to carry the products of as much as 28 machines (picture below). This is still a sketch and is not projected in the near future for VV, but this vehicle can be easily added in the MDMVRP model to conclude if it is profitable to invest in it or not.

Picture 6. New truck in consideration

3. Data sets description

A total of nine different classes of datasets were used along this project. Three of them rank as small instances which solely purpose is to check the correct execution of the method. These three instances does not have any utility to compare the efficiency of the method because of its size and were not replicated. Datasets four to eight are datasets of medium – large size. The purpose of these instances is to compare the efficiency of the traditional method with the column generation method. To avoid concluding based upon a single dataset, each of them were replicated three times with random data and the results are given using all of the replicas. Last but not least, dataset nine is a large scale problem and its purpose is to evaluate the column generation method in a situation where the simplex method is unpractical.

All the instances were created locating randomly the number of nodes in a Cartesian plane. Then the Euclidean distance was computed and the different cost matrixes were based upon it. The demand of the clients were generated randomly within some limits (it should not exceed the capacity of a van).

Table 1 provides a summary of all the instances:

Table 1. Summary of data sets

Data set	Problem	Nodes	Variables	Constraints	Replications
1	VRP	8	264	118	1
2	MDVRP	9	341	136	1
3	MDMVRP	9	341	136	1
4	VRP	13	1027	283	3
5	MDVRP	14	1202	310	3
6	MDMVRP	14	1202	310	3
7	MDMVRP	18	2300	487	3
8	MDMVRP	22	3910	704	3
9	MDMVRP	50	42584	3449	1

4. Column generation method description

Now, the algorithm used in this context is presented. It is important to remark that the algorithm described below corresponds to the MDMVRP which is the most complex. However, it is easy to extend this algorithm for the other models.

Column generation algorithm

Find a feasible solution *P1*
Add *P1* to *P'*
Set $SP_i = -1 \quad \forall i \in I$ where *I* is all the possible combinations of depots and type of vehicles.
Set *IterationsLimit*
Set *NumberOfIterations* to 1
While ($\exists\ SP_i > 0$ for any *i* or *NumberOfIterations* < *IterationsLimit*):
    Solve the linear relaxation of the RMP using paths *P'*
    For *i* from 1 to I
        If $SP_i < 0$
            Solve the SP
            Add solution P_i to *P'*
            Set SP_i to the optimal value of SP
            Set *NumberOfIterations* to *NumberOfIterations + 1*
        End If
    End For
End While
If *NumberOfIterations* ≥ *IterationsLimit*
    Display "Algorithm does not converge"
Else
    Solve the Integer RMP using paths *P'*
End If.

As for the initial feasible solution, a heuristic based upon the nearest neighbor algorithm was used. In this heuristic, each route starts in the depot and then the client with less cost associated is visited. This repeats until the vehicle does not have more capacity available to serve other clients so it returns to the depot.

5. Comparison of column generation with Simplex method

To compare the performance of the column generation algorithm with the simplex method in medium scale problems, the series of data sets described before were solved using both methods. The results are presented below in Table 2:

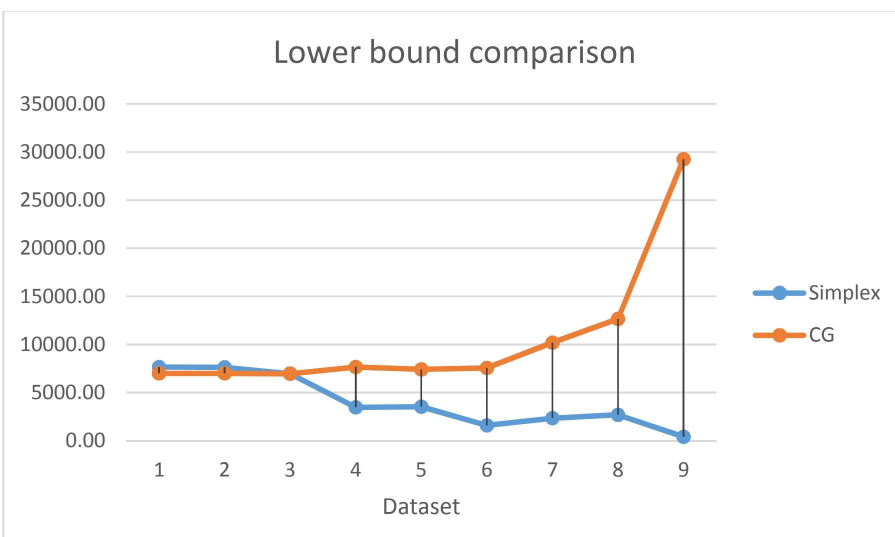
Table 2. Performance of column generation vs. Simplex method

Data set	Lower bound		Upper bound		Gap (%)		CPU (s)	
	Simplex	CG	Simplex	CG	Simplex	CG	Simplex	CG
1	7,663.35	7,009.56	7,663.35	7,669.12	0.00%	8.60%	0.5	1
2	7,644.23	7,009.56	7,644.23	7,669.12	0.00%	8.60%	0.5	1
3	6,981.85	6,981.85	6,981.85	6,981.85	0.00%	0.00%	0.5	1
4	3,469.18	7,656.62	8,629.09	8,750.13	60.36%	12.06%	3,600	16
5	3,541.33	7,407.10	8,586.07	8,646.94	59.35%	14.51%	3,600	58
6	1,594.82	7,567.11	8,133.35	8,124.65	80.47%	6.69%	3,600	109
7	2,357.59	10,202.60	10,636.85	10,482.64	78.02%	2.54%	3,600	301
8	2,685.56	12,653.17	13,273.81	13,098.24	79.68%	3.42%	3,600	504
9	413.00	29,230.40	32,120.00	29,814.40	98.71%	1.96%	4,800	8,676

Compared to the simplex method that is the standard solving method in Linear Programming, the column generation method had very interesting features that are important to explore.

5.1 The lower bound

The behavior of the lower bound in column generation represents a strong advantage over the lower bound that can be obtained using the simplex method. Excluding data sets one to three (due to the size of the network) the column generation algorithm outperformed the simplex method (as can be seen in Graphic 1) in all of the remaining data sets.

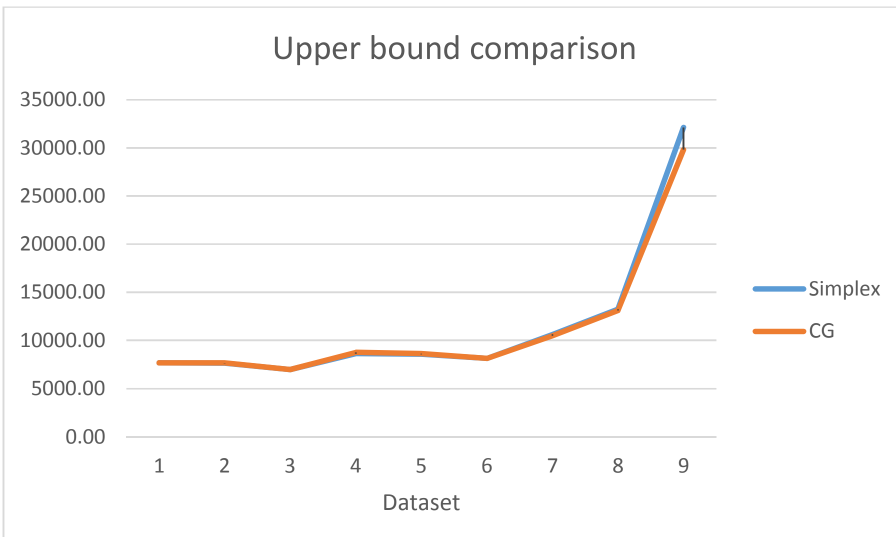


Graphic 1. Performance of the Lower bound in column generation vs. Simplex method

It is also important to remark that, as the number of nodes increase, the efficiency of the simplex method decreases compared to column generation. This gives a lot of importance to the property of the column generation method of using only a subset of available paths to solve *RMP* because allows the method to find better lower bounds.

5.2 The upper bound

As can be seen in Graphic 2, the behavior of the upper bound is practically equal in both methods. Nevertheless, there is one aspect not represented in the graphic that is very important to consider: the time. Although the results are very close, even sometimes the simplex method displays better ones, the time the simplex needs to achieve those results is much higher. Casting outside again data sets one to three, the simplex method had to be executed for an hour to obtain the results portrayed in Graphic 2, meanwhile the column generation method converged to a similar upper bound in less than ten minutes in the worst case. The only exception is data set nine (the large scale instance) but the performance this would be discussed later.

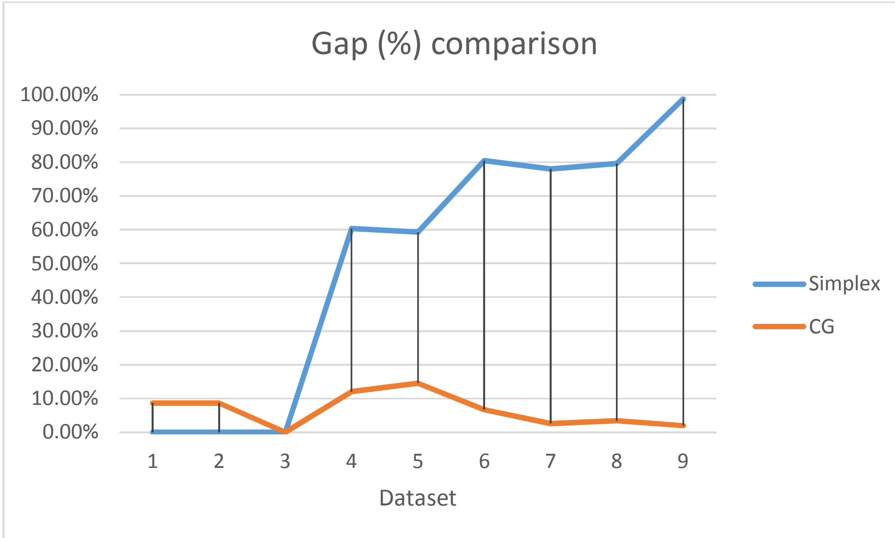


Graphic 2. Performance of the upper bound in column generation vs. Simplex method

5.3 The gap

As a direct consequence of the two previous indicators of performance of the methods, the gap is the most important differentiator between the simplex method and the column generation approach. As can be seen in Graphic 3 and using only data sets four to nine, difference in the gap increases gradually as the data set complexity increases too. The reason is the poor performance of the lower bound in the simplex method.

Considering the other key factor between methods, the time, the gap obtained using the column generation becomes even more important. The reason behind this analysis is that column generation allows the user to achieve a relatively small gap (and therefore know a decent range for the optimal solution of the *OP*) in a time that is not as prohibitive as achieving the same result using only the simplex method.



Graphic 3. Performance of the gap in column generation vs. Simplex method

6. Comparison of column generation results depending on the dataset

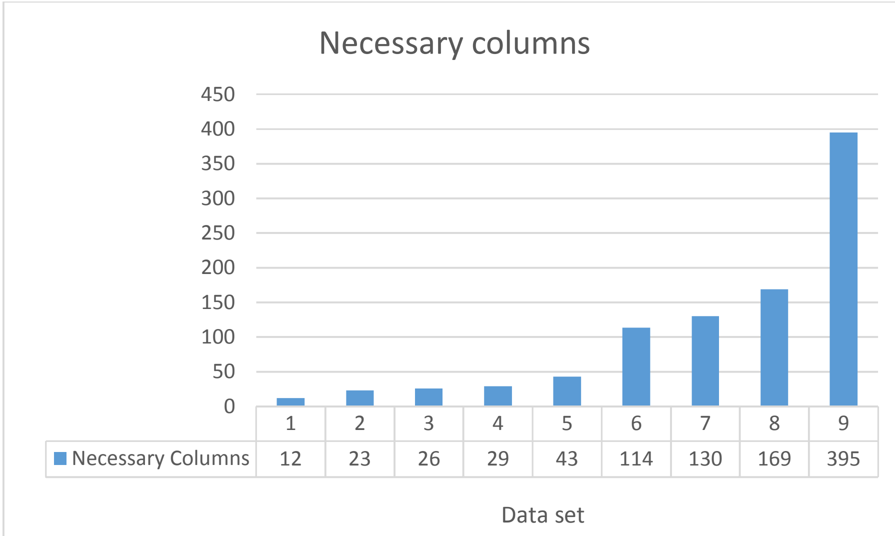
As each data set becomes more complex than the one before, a comparison between all the different set of information was conducted to understand the behavior of the column generation method. It is important to comment that the data sets with multiple replicas started with the same number of initial columns to eliminate a possible interaction of the initial solution with the final solution.

Table 3. Performance of column generation for different data sets

Data set	Nodes	Col_0	*#Col*	Gap	CPU (s)
1	8	4	12	8.60%	1
2	9	4	23	8.60%	1
3	9	4	26	0.00%	1
4	13	5	29	12.06%	16
5	14	5	43	14.51%	58
6	14	5	114	6.69%	109
7	18	7	130	2.54%	301
8	22	8	169	3.42%	504
9	50	17	395	1.96%	8,676

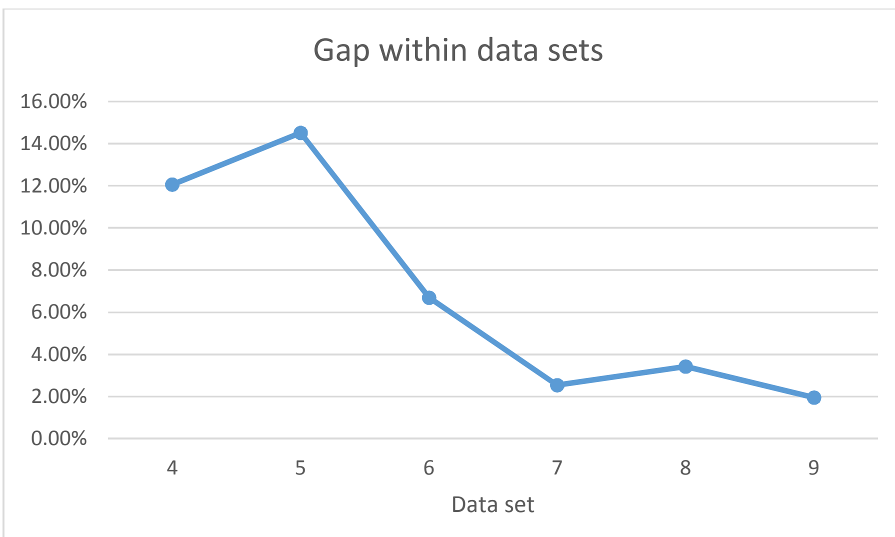
As complexity increases in the datasets solved, the behavior of the column generation method changes. There are two important factors that are directly attached to the complexity: time and number of columns needed to end the algorithm.

As can be seen in Table 3 and Graphic 4, a higher number of nodes, implies a higher number of columns needed and more computational processing time spent. The reason for the increasing of the number of columns to be generated is due to two factors: the first of them is the necessity to solve more *SP's* with the MDVRP and MDMVRP formulations. Thus, each of them requires new columns for each depot and in the second case, the MDMVRP, the number of new columns in every iteration is also based on the types of vehicles, a new column for each depot and vehicle type.



Graphic 4. Behavior of the necessary columns

The second reason is that increasing the network size also increases the number of paths available to fulfill the demand of the clients. Even though the RMP for the three formulation uses only a subset of the paths available, having a bigger network implies that more paths will have to be included in the RMP to find optimality and guarantee a lower bound to the OP.



Graphic 5. Behavior of the gap

Another important factor in the analysis of the performance of the method within the data sets is the gap. In the previous section the gap was considered the most useful indicator to evaluate how the method performs. In the case within different data sets the same result applies. As can be seen in Graphic 5 (in which only the data sets with replications are represented) the gap has a tendency to decrease with the complexity of the networks solved. The cause of this result is that a bigger network implies more paths added to the RMP as explained before. As a consequence, the method has a limited number of paths to compute a solution but all of them have in common that were generated with the solely purpose of improving the RMP. Therefore, the quality of paths available is very good and allows better results.

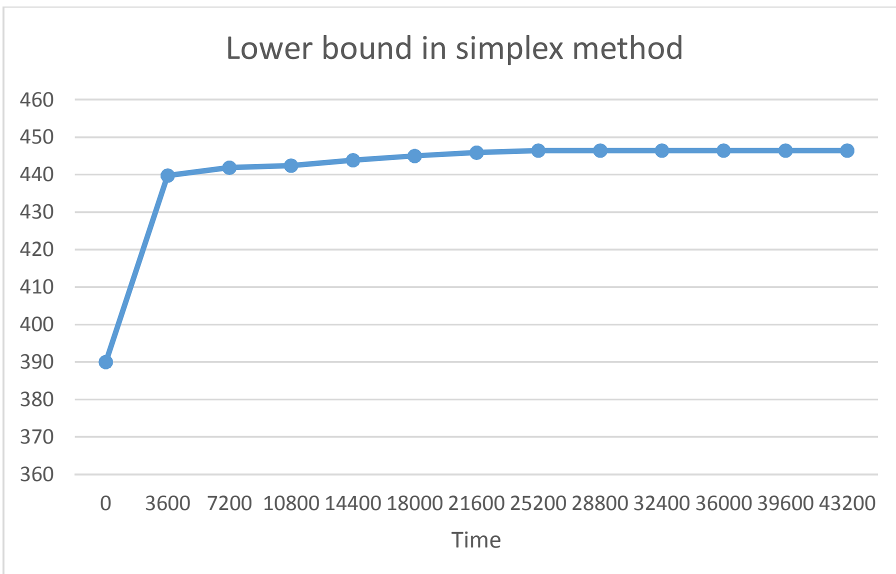
7. Performance of the column generation method in a large scale problem

With the purpose of evaluating how the column generation method would perform in a large scale problem, data set nine was used. This data set with two depots and 48 clients (total of 50 nodes) have an OP with more than 40,000 variables and 3,000 constraints. The simplex method algorithm was stopped after 43,200 seconds (12 hours).

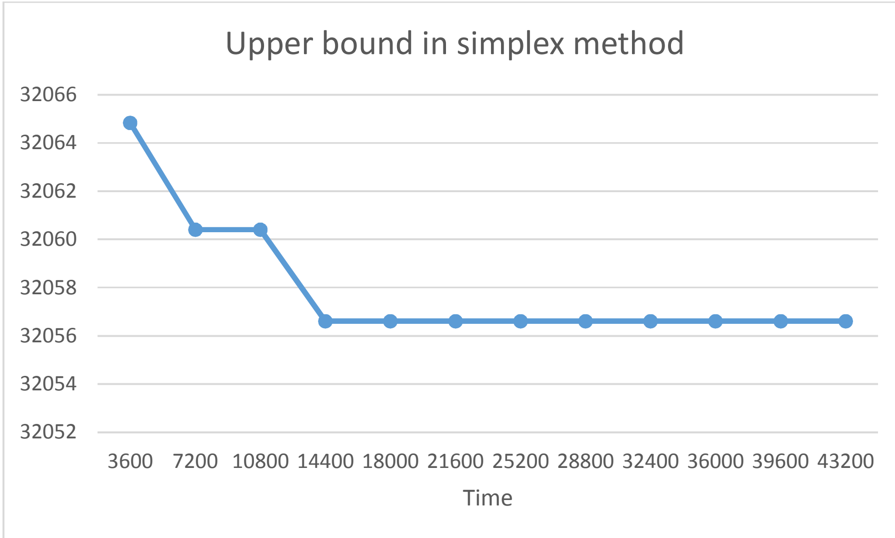
Table 4. Performance of simplex method compared to column generation

	Simplex	CG
Lower bound	446.41	29,230.40
Upper bound	32,056.61	29,814.4
Gap	98.61%	1.96%
CPU time (seconds)	43,200	8676
Root Node Linear Relaxation	390.00	29,230.40

The performance of the upper and lower bounds in time were evaluated in the simplex method and it can be seen in Graphic 6 and Graphic 7.

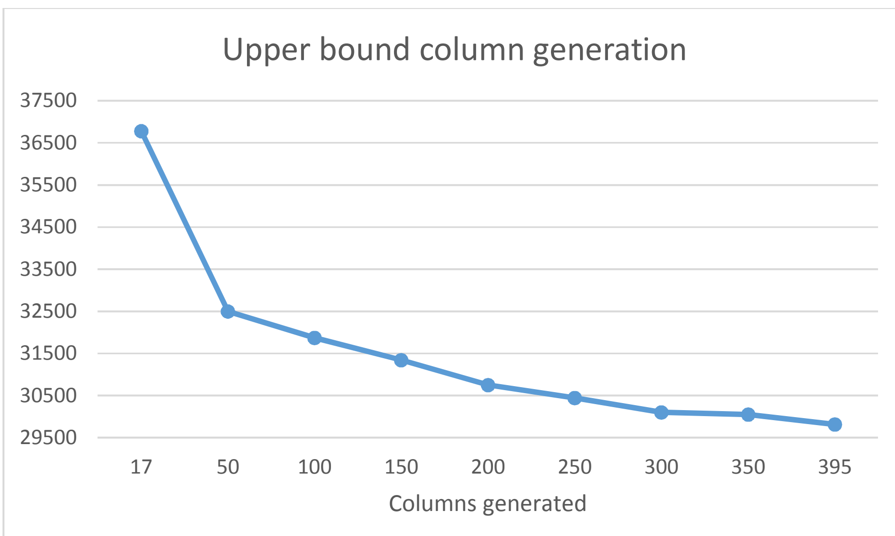


Graphic 6. Behavior of the lower bound in the simplex method



Graphic 7. Behavior of the upper bound in the simplex method

For the column generation approach, only the upper bound behavior was evaluated, in Graphic 8. The behavior of the lower bound in the column generation method is not presented as a result because the lower bound taken from solving the RMP_{MDMVRP} is not necessarily a valid lower bound for the OP. The reason is that RMP_{MDMVRP} only uses a subset of paths and not all of them.



Graphic 8. Behavior of the upper bound in the column generation method

To compare the behavior of column generation method in a large scale problem (data set nine) against the simplex method, the data set solution was executed for 12 straight hours (43.200 seconds) in the second method. As can be seen in Table 4, the column generation method clearly outperformed the simplex method in every indicator. Starting in the linear relaxation of the root node it can be seen that the simplex method's solution is far away from any feasible answer. On the contrary, the column generation method gives an answer much closer to an upper bound.

In defense to the simplex method, it can be stated that the linear relaxation of the root node can be obtained immediately meanwhile the same cannot be said for column generation. This is because the only valid lower bound in column generation can only be obtained after the algorithm converges. The problem with the simplex method is that the lower bound founded in the root node relaxation is very difficult to improve. As can be seen in Graphic 6, after 12 hours of execution, the lower bound only improved roughly 56 units (from 390 to 446). Furthermore, the worst performance of the lower bound occurs after the first hour where it takes 11 hours to improve in six units, from 440 to 446. On the contrary, after only approximately 2.5 hours, the Column Generation Method was able to find a unique lower bound more than 66 times bigger than the one computed by the simplex method in 12 hours. This result clearly demonstrates the usefulness of using only a subset of paths in the RMP. Due to this factor, every time the RMP is solved, only paths available interact with the solution, meanwhile in the simplex method all the paths interact without considering how unreasonable it is to use certain paths which are too expensive.

In the upper bound indicator, the simplex method initially is able to find a feasible solution of 32.065 improving it to 32,057 (approximately 8 units) in more than 12 hours. The column generation method starts with a feasible solution of 36,500 units, not that good. However after 100 columns and approximately 45 minutes it already improved the upper bound of the simplex method reaching an optimal solution of 29,814 when the algorithm converges. As a consequence, the difference between both bounds sits around 2,000 units. Again, this is an advantage of only using paths that can improve the RMP in every iteration casting outside expensive and inefficient paths.

The consequence of all the behaviors explained beforehand can be identified in the gap. Due to the lack of efficient lower and upper bounds improvements, after 12 hours the gap in the simplex method is still 98.6%. On the contrary, the gap in the Column Generation Algorithm is 1.96%.

Chapter 4

Conclusions

In the modern world where the limelight is frequently on the Logistics department, due to the necessity of doing more tasks with the same amount of money, approaching the VRP and its extensions with a systemic method is fundamental to achieve this goal. Being able to route an always expanding network of clients with different possibilities (multiple depots, multiple types of vehicles or both) in a close to optimum fashion is a competitive advantage that any Logistics department will have to learn.

Implementing the column generation method to solve the VRP, the MDVRP or the MDMVRP is an excellent option to be considered to obtain this competitive advantage. The method allows the user to obtain feasible routes to schedule the vehicles in the companies knowing a range in which the real minimum cost routing is located. This last characteristic can be used for those die-hard managers which doubt about the power of the column generation method against the empirical routing method. What can be done is a comparison between the solution that an experienced person can give compared with the one achieved using column generation. It is important to mention that the systemic solution can be obtained in a short span of time.

Another advantage of implementing the column generation method is that, in an always expanding network, adding a new client to the model doesn't change the formulation and the same algorithm can be used to find routes to serve the client. Meanwhile, in an empirical solution, adding clients mean using intuition every time to get a solution. The problem is that intuition is not always trustworthy.

Now, the question is how good is the method of column generation not against intuition but against the simplex method that is the standard method to solve this kind of problems. The key factor in this comparison resides in the property of the column generation method that allows to obtain a lower and upper bound without the necessity of using all the variables but a subset o them. This characteristic has an implication in the computational processing time because limits every iteration of the RMP to a very small fraction of variables compared to the original MP.

A lot of consequences derive from this computational processing time. The first of them is that it allows the method to converge much faster than the simplex method. In fact, after increasing the number of clients from seven to 12, the simplex methods stops giving usable bounds in an hour of processing time. On the other hand, the column generation method gives much better bounds and in less time.

Another advantage obtained using column generation (and partially still a consequence of the computational processing time) is the gap. Due to the slow convergence of the simplex method, it performs poorly in obtaining a good gap. The problem in which this can be easily seen is in the large scale problem. After more than 12 hours of computational processing time, the gap found using the simplex method was a terrible 98.6%. And the 1.96% gap achieved using column generation makes the gap obtained through simplex method look even worse.

This difference in the gap performance have two different causes: the first of them, as commented above, is the computational processing time which slows the improvement of the lower and upper bound. But the second, which hasn't been discussed yet, is the stronger linear relaxation of the RMP using column generation. This stronger linear relaxation is obtained because of the use of only a subsets of variables instead of all of them. As such, the algorithm has to find the best linear relaxation using the best solutions obtained previously based in the dual variables. Then, when the integer RMP is solved, the difference between the upper bound and the lower bound is very decent.

Last but not least, the algorithm presents a logical behavior when compared between different data sets. There is a directly proportional ratio between the complexity of the problem and the number of columns that has to be generated for the algorithm to converge. Given the fact that in all the data sets the initial columns were generated using the same method, the behavior of the algorithm using different approaches for the initial columns is left as future research.

Bibliography

Choi, E. Tcha, D. (2007). A Column Generation Approach to the Heterogeneous Fleet Vehicle Routing Problem. *Computers & Operations Research, 34, 2080-2095*.

Desrosiers, J. Lübbecke, M. (2005). A Primer in Column Generation. *Column Generation, 1-29.*

Desrosiers, J. Lübbecke, M. (2005). Selected Topics in Column Generation. *Operations Reasearch, 53, 1007-1023.*

Feillet, D. (2010). A Tutorial on Column Generation and Branch-and-price for Vehicle Routing Problems. *4OR, 4, 407-424.*

Kallehauge, B. Larsen, J. Madsen, O. Solomon, M. (2005). Vehicle Routing Problem with Time Windows. *Column Generation, 67-97.*

Lübbecke, M. (2010). Column Generation. *Wiley Encyclopedia of Operations Research and Management Science*.

Posada, J. Farbiarz, V. González, C. (2011). Análisis del "Pico y Placa" como Restricción a la Circulación Vehicular en Medellín – Basado en Volúmenes Vehiculares. *Revista DYNA, 78, 112-121.*

Nazareth, J. (1984). Numerical Behaviour of LP Algorithms Based Upon the Decomposition Principle. *Linear Algebra Appl, 57, 181-189.*

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. Jhon Wiley and Sons, Chichester, UK.