# III
# Mathematical Formulations

This chapter presents the formulations used in this work. Firstly, we show the set partitioning approach, which can be applied to most of known routing problems. Next, for each problem, we show some specific existing formulations and how to transform them to obtain the set partitioning formulation.

## III.1 The Set Partitioning Approach

The Set Partitioning Approach is a simple way to formulate VRPs. In this formulation, a binary variable $\lambda_r$ is created for each route. As mentioned before, a route is a walk which starts at a depot, services a subset of the customers and returns to the same depot. Given the set of all possible routes, called $\Omega$, the objective is to select exactly $|\mathcal{K}|$ routes, where all customers are serviced and total cost is minimized. This formulation is presented below.

$$\text{Minimize } \sum_{r \in \Omega} c_r \lambda_r \tag{III.1}$$

$$\text{s.t.} \quad \sum_{r \in \Omega} \lambda_r = |\mathcal{K}| \tag{III.2}$$

$$\sum_{r \in \Omega} a_r^i \lambda_r = 1 \quad \forall i \in \mathcal{C} \tag{III.3}$$

$$\lambda_r \in \{0, 1\} \quad \forall r \in \Omega \; . \tag{III.4}$$

The objective function (III.1) minimizes the overall cost of the selected routes, where $c_r$ is the cost of route $r$. Constraints (III.2) impose the number $|\mathcal{K}|$ of vehicles to be used. Given the binary constant $a_r^i$, which indicates whether customer $i$ is serviced by route $r$, constraints (III.3) guarantee that each customer is serviced by exactly one route. Finally, constraints (III.4) force the variables to be binary.

It is easy to see that $\Omega$ contains an exponential number of routes, thus evolving at an exponential number of variables. This is the drawback of this formulation, as it is necessary to use column generation in order to deal with the large number of variables involved. Anyway, the quality of the solutions obtained using this approach is very good, what can be evidenced by the existence of several works using this formulation to solve VRPs.

Ideally, it is preferable that the set $\Omega$ contains only feasible routes, called elementary routes. However, as we will show later on, this restriction turns the pricing subproblem into a problem that is hard to solve. A way to handle this difficulty is to relax the elementarity of the routes, allowing them to service same customer more than once. This kind of non-elementary route is also known as the *q-routes*. Since pricing non-elementary routes is less difficult than pricing elementary routes, it is expected as a side effect a reduction in the value of the bounds obtained. For this reason, recent works try to find a compromise between elementary and non-elementary routes, restricting the non-elementarity in different ways.

## III.2 Formulations for the CARP

### (a) The Two-Index Formulation

An intuitive formulation for the CARP is the *Two-Index Formulation*. According to Letchford and Oukil [53], which was also the first work to use this name to the formulation, this approach was first suggested by Welz [76] in 1994, but it was only explored in depth in the work of Belenguer and Benavent [11] in 1998. In this formulation, there is a binary variable $x_e^k$, for each required edge and each vehicle, which is 1 if a vehicle $k$ services the required edge $e$ and 0 otherwise. Furthermore, there is an integer variable $z_e^k$, for every edge and each vehicle, which represents the number of times vehicle $k$ deadheads edge $e$. These variables are used as follows.

$$\text{Minimize} \sum_{k \in \mathcal{K}} \left( \sum_{e \in E_R} c_e x_e^k + \sum_{e \in E} c_e z_e^k \right) \tag{III.5}$$

subject to

$$\sum_{k \in \mathcal{K}} x_e^k = 1 \qquad \forall e \in E_R \tag{III.6}$$

$$\sum_{e \in E_R} d_e x_e^k \leq Q \qquad \forall k \in \mathcal{K} \tag{III.7}$$

$$\sum_{e \in \delta_R(S)} x_e^k + \sum_{e \in \delta(S)} z_e^k \geq 2x_f^k \qquad \forall S \subseteq V \backslash \{0\}, f \in E_R(S), k \in \mathcal{K} \tag{III.8}$$

$$\sum_{e \in \delta_R(S)} x_e^k + \sum_{e \in \delta(S)} z_e^k \equiv 0 \bmod (2) \qquad \forall S \subseteq V \backslash \{0\}, k \in \mathcal{K} \tag{III.9}$$

$$x_e^k \in \{0, 1\} \qquad \forall e \in E_R, k \in \mathcal{K} \tag{III.10}$$

$$z_e^k \in \mathbb{Z}_0^+ \qquad \forall e \in E, k \in \mathcal{K} \ . \tag{III.11}$$

The objective function (III.5) minimizes the overall cost of the edges used. Constraints (III.6) assure that all required edges are serviced. Constraints (III.7) limit the total demand serviced by each vehicle to the capacity $Q$. Given a vertex set $S$, let $\delta(S) = \{(i,j) \in E : i \in S \wedge j \notin S\}$ be the set of edges which have one endpoint inside $S$ and the other outside $S$ and let $\delta_R(S) = \{(i,j) \in E_R : i \in S \wedge j \notin S\}$ be the set of *required* edges which have one endpoint inside $S$ and the other outside $S$. Analogously, define $E(S) = \{(i,j) \in E : i \in S \wedge j \in S\}$ and $E_R(S) = \{(i,j) \in E_R : i \in S \wedge j \in S\}$ as the sets of edges with both endpoints inside $S$. Constraints (III.8) assure that every route is connected and constraints (III.9) force every route in the solution to induce an Eulerian graph. Notice that these latter constraints can be easily modeled as MIP constraints by introducing an integer variable which should appear on the right-hand side with a coefficient of 2.

The drawback of this formulation comes from its large number of variables and its symmetry. Letchford and Oukil [53] notice that there are at least $|\mathcal{K}|!$ optimal solutions. These turn this formulation prohibitive to be solved for instances with a large number of vehicles. The work of Ghiani et al. [50] shows how this symmetry can be tackled in order to use the formulation in a branch-and-cut algorithm.

## (b) The One-Index Formulation

In their work of 2003, Belenguer and Benavent [10] developed another CARP formulation, usually referred as the *One-Index Formulation* [53]. In contrast to other approaches, this formulation only makes use of variables representing the deadheading of an edge. In addition, all vehicles are aggregated. Due to these simplifications, this formulation is a relaxation, i.e., it may result in an infeasible solution for the original problem. Moreover, it is strongly $\mathcal{NP}$-hard to decide whether a solution given by this formulation is feasible or not. Nevertheless, these issues do not prevent such formulation from giving very good lower bounds in practice.

For each deadheaded edge $e$, there is an integer variable $z_e$ representing the number of times edge $e$ is deadheaded by *any* vehicle. Given a vertex set $S$, with $|\delta_R(S)|$ odd, it is easy to conclude that at least one edge in $\delta(S)$ must be deadheaded because each vehicle entering the set $S$ must leave and return to the depot. This is the definition of the following *odd-edge cutset inequalities*.

$$\sum_{e \in \delta(S)} z_e \geq 1 \quad \forall S \subseteq V \backslash \{0\}, |\delta_R(S)| \text{ odd} \tag{III.12}$$

Furthermore, given a lower bound on the number of vehicles needed to meet the demands in $\delta_R(S) \cup E_R(S)$, called $k(S)$, we can state that at least $k(S)$ vehicles must enter and leave the set $S$. Thus, at least $2k(S) - |\delta_R(S)|$ times an edge in $\delta(S)$ will be deadheaded. If this value is positive, we can define the following *capacity cut*.

$$\sum_{e \in \delta(S)} z_e \geq 2k(S) - |\delta_R(S)| \quad \forall S \subseteq V \backslash \{0\} \tag{III.13}$$

The best value for $k(S)$ is obtained by solving the Bin Packing problem for each vertex set $S$, but as shown by Garey and Johnson [32], this problem is an $\mathcal{NP}$-hard problem. Due to this fact, we prefer to use the approximation $k(S) = \lceil \sum_{e \in \delta_R(S) \cup E_R(S)} d_e / Q \rceil$.

Since the left-hand side of both (III.12) and (III.13) are the same, they can be represented in the formulation by using just a single constraint. This can be done by introducing $\alpha(S)$, which is defined as follows.

$$\alpha(S) = \begin{cases} \max\{2k(S) - |\delta_R(S)|, 1\}, \text{ if } |\delta_R(S)| \text{ is odd,} \\ \max\{2k(S) - |\delta_R(S)|, 0\}, \text{ if } |\delta_R(S)| \text{ is even.} \end{cases} \tag{III.14}$$

The odd-edge cutset inequalities and the capacity cuts are the only constraints of the one-index formulation, besides the limits on the $z_e$ variables. Therefore, the formulation can be defined as follows.

$$\text{Minimize} \sum_{e \in E} c_e z_e \tag{III.15}$$

subject to

$$\sum_{e \in \delta(S)} z_e \geq \alpha(S) \quad \forall S \subseteq V \backslash \{0\} \tag{III.16}$$

$$z_e \in \mathbb{Z}_0^+ \qquad \forall e \in E \tag{III.17}$$

The objective function (III.15) minimizes the cost of the deadheaded edges. Constraints (III.16) combine cuts (III.12) and (III.13). It is important to notice that in order to obtain the overall cost of a solution, it is necessary to add the costs of the required edges ($\sum_{e \in E_R} c_e$) to the resulting cost.

Notice that, by the definition of the set $S$, it is easy to see that there is an exponential number of odd-edge cutset inequalities and capacity cuts. For this reason, it is not an option to generate all these cuts *a priori*. These must be generated in a cutting plane approach using a separation routine, as the one to be presented in Chapter IV.

## (c) The Set Partitioning Approach

The set partitioning formulation for the CARP can be obtained from the two-index formulation with the introduction of the binary variable $\lambda_r$ for every possible CARP route. A CARP route is a walk which starts at the depot, traverses a set of required edges – servicing their demands – and then returns to the depot without exceeding the capacity of the vehicle. Between a pair of serviced edges, the route may traverses a set of deadheaded edges, which contribute for the total cost of the route, but not for the total capacity.

The new variables $\lambda_r$ have a natural association with the $x_e^k$ and $z_e^k$ variables from the two-index formulation. Given the binary constant $a_r^e$, which is 1 if and only if route $r$ services the required edge $e$ and the integer constant $b_r^e$, which represents the number of times edge $e$ is deadheaded in route $r$, the two-index formulation can be rewritten as follows.

$$\text{Minimize} \sum_{k \in \mathcal{K}} \left( \sum_{e \in E_R} c_e x_e^k + \sum_{e \in E} c_e z_e^k \right) \tag{III.18}$$

subject to

$$\sum_{r \in \Omega} a_r^e \lambda_r = \sum_{k \in \mathcal{K}} x_e^k \quad \forall e \in E_R \tag{III.19}$$

$$\sum_{r \in \Omega} b_r^e \lambda_r = \sum_{k \in \mathcal{K}} z_e^k \quad \forall e \in E \tag{III.20}$$

$$\sum_{r \in \Omega} \lambda_r = |\mathcal{K}| \tag{III.21}$$

$$\sum_{k \in \mathcal{K}} x_e^k = 1 \quad \forall e \in E_R \tag{III.22}$$

$$\lambda_r \in \{0,1\} \quad \forall \lambda \in \Omega \tag{III.23}$$

$$x_e^k \in \{0,1\} \quad \forall e \in E_R, k \in \mathcal{K} \tag{III.24}$$

$$z_e^k \in \mathbb{Z}_0^+ \quad \forall e \in E, k \in \mathcal{K}\ . \tag{III.25}$$

Constraints (III.19) and (III.20) create the relation between variables from the two-index formulation and the $\lambda_r$ variables. Furthermore, due to the definition of a feasible CARP route and the introduction of the new variables, constraints (III.7), (III.8) and (III.9) are no longer required. Besides that, it is important to notice that due to the aggregation of the vehicles, constraint (III.21) is introduced in order to force the given number of vehicles to be used.

Variables $x_e^k$ and $z_e^k$ can be completely replaced using the relation constraints (III.19) and (III.20). This variable replacement and the relaxation of the integrality constraints (III.23) generate the following Dantzig-Wolfe relaxation.

$$\text{Minimize} \sum_{r \in \Omega} c_r \lambda_r \tag{III.26}$$

subject to

$$\sum_{r \in \Omega} \lambda_r = |\mathcal{K}| \tag{III.27}$$

$$\sum_{r \in \Omega} a_r^e \lambda_r = 1 \qquad \forall e \in E_R \tag{III.28}$$

$$\lambda_r \in [0, 1] \quad \forall \lambda \in \Omega \tag{III.29}$$

$$\tag{III.30}$$

The objective function (III.26) minimizes the total cost of the used routes, where the cost of routes is given by $c_r = \sum_{e \in E_R} c_e a_r^e + \sum_{e \in E} c_e b_r^e$. Constraints (III.27) limit the number of routes used to the number of available vehicles and constraints (III.28) assure each required edge is serviced by exactly one route.

Furthermore, in order to improve the bounds of this linear relaxation, the odd-edge cutset inequalities and the capacity cuts can be also rewritten using the $\lambda_r$ variable and included in this formulation. The resulting constraint from this modification applied on (III.16) is as follows.

$$\sum_{r \in \Omega} \sum_{e \in \delta(S)} b_r^e \lambda_r \geq \alpha(S) \quad \forall S \subseteq V \backslash \{0\} \tag{III.31}$$

# III.3 Formulations for the GVRP

## (a) The Undirected Formulation with an Exponential Number of Constraints

The *Undirected Formulation with an Exponential Number of Constraints* was proposed by Bektaş et al. [8] and uses one type of variable, called $z_e$, which represents the number of times any vehicle traverses edge $e$. In general, these variables are binary variables, i.e., they can hold only values 0 or 1. The only exception occurs when the edge $e$ is adjacent to the deposit, in this case they can hold a third value. When a vehicle leaves the depot, services just one cluster and then returns to the depot using the same edge, the variable receives the value 2.

$$\text{Minimize} \sum_{e \in E} c_e z_e \qquad \text{(III.32)}$$

subject to

$$\sum_{e \in \delta(C_0)} z_e = 2|\mathcal{K}| \qquad \text{(III.33)}$$

$$\sum_{e \in \delta(C_m)} z_e = 2 \qquad \forall m \in M \backslash \{0\} \qquad \text{(III.34)}$$

$$\sum_{e \in \delta(S)} z_e + 2 \sum_{(i,j) \in L : i \notin S} z(\{i\} : C_j) \leq 2 \qquad \forall m \in M \backslash \{0\}, \forall S \subseteq C_m, \forall L \in \bar{L}_m \qquad \text{(III.35)}$$

$$\sum_{e \in \delta(S)} z_e \geq 2k(S) \qquad S \subseteq C \backslash \{C_0\} \qquad \text{(III.36)}$$

$$z_e \in \{0, 1, 2\} \qquad \forall e \in \delta(0) \qquad \text{(III.37)}$$

$$z_e \in \{0, 1\} \qquad \forall e \in E \backslash \delta(0) \qquad \text{(III.38)}$$

The objective function (III.32) minimizes the total traversal cost. Constraints (III.33) ensure that the degree of the depot cluster is $2|\mathcal{K}|$, forcing the number of vehicles to be equal to $|\mathcal{K}|$. Constraints (III.34) ensure that the degree of any cluster $m \neq 0$ is 2, forcing all these clusters to be serviced by only one vehicle.

Constraints (III.35) are the *same-vertex inequalities*. Given $\bar{L}_m = \{L : L \subseteq \bigcup_{i \in C_m} L_i, |L \cap L_i| = 1, \forall i \in C_m\}$ where $L_i = \{i\} \times (M \backslash \{0, \mu(i)\})$ for all $i \in V \backslash \{0\}$, they ensure that if a vehicle enters in a cluster using a vertex $v$, it must leave the cluster using the same vertex. We refer the reader to the work of Bektaş et al. [8], where an extensive explanation of the same-vertex inequalities can be found, together with a graphic example, the proof of its correctness and a separation routine.

Constraints (III.36) are the *capacity constraints*, as they ensure that in the solution there will be no vehicle violating the capacity $Q$ and there will be no subtours, i.e., every route must be connected with the depot. Analogously to the CARP, given a cluster set $S$, $k(S)$ represents a lower bound on the number of vehicles needed to service the cluster set $S$. As mentioned before, this can be calculated by solving the Bin Packing problem, which is $\mathcal{NP}$-hard [32]. However, in the GVRP case the approximation used is $k(S) = \lceil \sum_{m \in S} d_m / Q \rceil$.

Notice that, as well as the CARP capacity cuts, there is an exponential number of GVRP capacity cuts and therefore *a priori* generation may not be possible. So, a cutting plane algorithm during the resolution of the problem is required.

## (b) Set Partitioning Approach

The undirected formulation with an exponential number of constraints can be naturally rewritten using route variables. A GVRP route is a walk that starts at the depot, traverses a set of clusters without violating the vehicle capacity $Q$, and returns to the depot. Given the set of all GVRP routes, called $\Omega$. We define a binary variable $\lambda_r$, $\forall r \in \Omega$, which is 1 when the route $r$ is used by a vehicle in the solution, and 0 otherwise. Let $b_r^e$ be a constant value indicating how many times route $r$ traverses edge $e$. Therefore, the new formulation is as follows.

$$\text{Minimize} \sum_{e \in E} c_e z_e \tag{III.39}$$

subject to

$$\sum_{r \in \Omega} b_r^e \lambda_r = z_e \qquad \forall e \in E \tag{III.40}$$

$$\sum_{e \in \delta(C_0)} z_e = 2|\mathcal{K}| \tag{III.41}$$

$$\sum_{e \in \delta(C_m)} z_e = 2 \qquad \forall m \in M \backslash \{0\} \tag{III.42}$$

$$\lambda_r \in \{0,1\} \qquad \forall r \in \Omega \tag{III.43}$$

$$z_e \in \{0,1,2\} \qquad \forall e \in \delta(0) \tag{III.44}$$

$$z_e \in \{0,1\} \qquad \forall e \in E \backslash \delta(0) \tag{III.45}$$

Constraints (III.40) define the relation between variables $z_e$ and $\lambda_r$. Notice that constraints (III.35) and (III.36) are no longer required because of the definition of the GVRP routes together with constraints (III.40).

From now on, we can replace the $z_e$ variables using its relation with the $\lambda_r$ variables and relax the integrality constraints (III.43) to obtain the Dantzig-Wolfe relaxation as follows.

$$\text{Minimize} \sum_{r \in \Omega} \left( \sum_{e \in E} c_e b_r^e \right) \lambda_r \tag{III.46}$$

subject to

$$\sum_{r \in \Omega} \left( \sum_{e \in \delta(C_0)} b_r^e \right) \lambda_r = 2|\mathcal{K}| \tag{III.47}$$

$$\sum_{r \in \Omega} \left( \sum_{e \in \delta(C_m)} b_r^e \right) \lambda_r = 2 \qquad \forall m \in M \backslash \{0\} \tag{III.48}$$

$$\lambda_r \in [0, 1] \quad \forall r \in \Omega \tag{III.49}$$

$$\tag{III.50}$$

There is a simpler way to write the above formulation. Knowing that every route must start and end at the depot, we can state that $\sum_{e \in \delta(C_0)} b_r^e = 2$, $\forall r \in \Omega$. Given $a_r^m \in \{0, 1\}$, the number of times route $r$ traverses cluster $m$, by the definition of $b_r^e$, we can also state that $\sum_{e \in \delta(C_m)} b_r^e = 2a_r^m$, $\forall m \in M \backslash \{0\}$ and $\forall r \in \Omega$. Furthermore, we can define the total cost of a route $r$ as $c_r = \sum_{e \in E} c_e b_r^e$, $\forall r \in \Omega$. Therefore, the final formulation follows:

$$\text{Minimize} \sum_{r \in \Omega} c_r \lambda_r \tag{III.51}$$

subject to

$$\sum_{r \in \Omega} \lambda_r = |\mathcal{K}| \tag{III.52}$$

$$\sum_{r \in \Omega} a_r^m \lambda_r = 1 \qquad \forall m \in M \backslash \{0\} \tag{III.53}$$

$$\lambda_r \in [0, 1] \quad \forall r \in \Omega \tag{III.54}$$

Notice that the simplicity of this formulation makes it easy to understand that constraints (III.52) ensure that $|\mathcal{K}|$ vehicles must be used and constraints (III.53) ensure that every cluster must be visited only once.

In order to improve the solution obtained by this relaxation, we can define some cuts to be used within the formulation. Using the equation (III.40), any cut which is valid for the undirected formulation can be translated into a valid cut for the set partitioning formulation. Thus, we can rewrite the capacity cuts (III.36) as follows.

$$\sum_{r \in \Omega} \sum_{e \in \delta(S)} b_r^e \lambda_r \geq 2k(S) \quad S \subseteq C \backslash \{C_0\} \tag{III.55}$$