

A priori performance measures for arc-based formulations of the Vehicle Routing Problem*

Fernando Ordóñez,[†] Ilgaz Sungur, and Maged Dessouky
Industrial and Systems Engineering, University of Southern California
3715 McClintock Ave, GER 240, Los Angeles, CA 90089
{fordon,sungur,maged}@usc.edu

July 2006

Abstract

The Vehicle Routing Problem (VRP) is a central problem for many transportation applications, and although it is well known that it is difficult to solve, how much of this difficulty is due to the formulation of the problem is less understood. In this paper we experimentally investigate how the solution times to solve a VRP with a general IP solver are affected by the formulation of the VRP used. The different formulations are evaluated by examining solution efficiency as a function of several a priori performance measures based on the data parameters. Our experimental results show how the solution run times are sensitive to problem parameters; in particular the sensitivity of formulations to the coefficient of variation of the cost matrix of travel times is explained by two interacting factors.

*Research supported by NSF under grant CMS-0409887

[†]Corresponding author

Keywords: Vehicle routing; arc-based formulation; computation; performance measures

1 Introduction

Many industrial applications deal with the problem of routing a fleet of vehicles from a depot to service at minimum cost a set of customers that are geographically dispersed. This type of problem can be formulated as The Vehicle Routing Problem (VRP) and it is faced daily by courier services (e.g., Federal Express, United Parcel Service, and Overnight United States Postal Service), local trucking companies, and demand responsive transportation services, just to name a few. These types of services have experienced tremendous growth in recent years. For example, both United Parcel Service and Federal Express have annual revenue of well over \$10 billion, and the dial-a-ride service for the disabled and handicapped is today a \$1.2 billion industry (Palmer, Dessouky, and Abdelmaguid 2004). Therefore, there is an increasing need to develop efficient routing and scheduling tools for the VRP. There exists a substantial amount of research on applications, formulations, and solution approaches for this problem; see Laporte (1992), Toth and Vigo (2002a), and Toth and Vigo (2002b) for an overview of the VRP.

When dealing with the complexity and practical difficulty of solving a problem, three aspects of the problem become relevant: the problem size, how the problem is formulated, and the algorithm used to solve the problem. In fact, classic complexity theory studies the asymptotic behavior of algorithms as the problem size increases, and on the practical side, a number of algorithms are developed to address specific formulations of the problem. In particular, the VRP is known to be NP-hard (Lenstra and Rinnooy Kan 1981), and as we discuss in more detail later, there are specialized algorithms for the VRP in its path

variable formulation (column generation) or arc variable formulation (branch-and-bound). It is well known that highly constrained VRPs, such as those with tight capacity or time window constraints, are more amenable to a column generation solution approach since the constraints leads to pruning up front a large number of infeasible paths (Lu and Dessouky 2004). Therefore, there is clearly a strong relationship between specific algorithms and problem formulation. However, little is known about how the formulation of the problem influences the intrinsic difficulty of the problem and hence the solution run time of any algorithm for VRPs of a fixed size in practice.

In this paper we aim to study experimentally the effect of problem formulation on solution run times for the VRP when solved optimally by a commercial integer programming (IP) solver. The goal of this work then is similar to Jones et al. (1993), which investigates the effect of problem formulation of multicommodity flow problems on decomposition algorithms. However, by considering a general purpose solver we aim to identify problem features that can indicate whether a problem formulation is fundamentally easier to solve or not. Therefore, special solution procedures favoring particular formulations are avoided in this study. We note that there is a vast literature of experimental analysis on well-known hard VRP instances combining formulation and solution procedure. Some recent examples are Baldacci et al. (2004), Ralphs et al. (2003), Ralphs (2003). Since such studies develop special solution procedures for a given formulation, they do not identify the sole effect of the problem formulation independent from its special solution procedure.

When routing a fleet of vehicles, being able to identify a VRP formulation that is easier to solve can provide better routing solutions through improved solution methods that use the correct formulations. The appropriate VRP formulations can improve solution methods because of two different reasons. First, it can help steer algorithm research to focus on “good” formulations that are amenable to fast solutions. The idea being that research on these fundamentally simpler problem formulations can lead to improved algorithms. Sec-

ond, general purpose solvers such as the one used in this study have improved considerably over time. These solvers can now be used to obtain exact solutions to small problems and approximate solutions for large sized problems by stopping the execution after a run time limit. Therefore for problems of suitable size, “good” formulations can be identified based on problem instances to obtain solutions in a reasonable time.

In this study the effect of problem formulation is investigated based on a priori performance measures. We are interested in identifying measures based on problem data to describe the difficulty of the problem instance due to problem formulation, without solving the problem. There are other performance measures proposed in the VRP literature to investigate the effect of problem formulation. However, these measures are mainly posterior and typically depend on the optimal solution. For example the average number of customers per route and the number of arcs selected in all optimal solutions also inform how difficult it is to solve problems. In this study, we experimentally investigate the relation between problem data and problem formulation on the difficulty faced by a general IP solver in solving a VRP.

We consider the capacitated asymmetric vehicle routing problem with unit demand at each demand location. Our study concentrates on three different arc-based formulations of the VRP and uses CPLEX 8.1 as our state-of-the-art commercial IP solver to obtain exact solutions. We limit our study to arc-based formulations because path-based formulations of the VRP have an additional difficulty due to the variable explosion and thus require specialized column generation algorithms. The experiments conducted explore whether vehicle capacity, number of vehicles, and properties of the matrix of travel times, such as mean travel time or number of triangular inequalities violated, influence solution run times. We present plots comparing the effect on the run times of the different formulations by modifying each of these problem parameters.

In the next section we review different methods to solve VRPs and research on performance measures. Section 3 discusses different formulations of the VRP and introduces the problems considered in this study. We present our experimental results in Section 4 and finish the paper with conclusions and general remarks in Section 5.

2 Solving VRPs and Performance Measures

In this review section we discuss the different exact and approximate solution procedures that exist for the VRP and cover different performance measures for solution algorithms that have been considered in related contexts.

Our discussion of solution methods for the capacitated VRP follows the review presented in Toth and Vigo (2002b), which includes both exact and heuristic solution methods. Exact solution methods can be classified as branch-and-bound algorithms (Christofides et al. 1981; Laporte and Norbert 1987) and branch-and-cut algorithms (Ralphs et al. 2003) that solve arc-based vehicle or commodity flow models of the VRP, and column generation methods that solve path-based (or set covering) formulations of the VRP (Agarwal et al. 1989; Hadjiconstantinou et al. 1995). Research on these algorithms includes the development of good methods to tighten the relaxations of the integer problem in order to shrink the search tree. Some recent examples include Martinhon et al. (2004) which use a k -degree center tree-based relaxations and Baldacci et al. (2004) which develop a strengthened LP relaxation of a new problem formulation. In addition there is research on specializing these methods for specific variants of the VRP, such as Desrochers et al. (1992) which use a column generation method for the VRP with time windows or Lu and Dessouky (2004) which use a branch-and-cut approach for the VRP with pickup and delivery. The efficient solution of VRPs is still a challenging problem, which requires the use of

heuristics for solutions to large problems. Cordeau et al. (2002) provide a good contemporary survey of heuristic methods. There are two well-known classic heuristics: the savings heuristic (Clarke and Wright 1964) and the sweep algorithm (Gillett and Miller 1974). Other popular approximate methods are meta-heuristics such as tabu search (Gendreau et al. 1994), genetic algorithms (Potvin and Bengio 1996), simulated annealing (Osman 1993), and evolutionary algorithms (Hombberger and Gehring 1999).

Algorithms, both exact and approximate, are evaluated by their performance on standard suites of test problems, available for example from the webpage by Dorronsoro Díaz (2005). The difficult nature of the VRP is evident in the performance achieved by the algorithms. For example, the recent algorithm by Baldacci et al. (2004) solved problem **F-n135-k7** with 134 customers and 7 vehicles (Fisher 1994), reportedly the largest problem solved to optimality in the literature at the time. However, this algorithm was unable to solve to optimality some smaller instances, such as **E-n76-k7** (Christofides and Eilon 1969), with 75 customers and 7 vehicles. An observation from these test suites is that some solution methods perform well on certain instances while not on others. Another complicating factor is that some solution methods require additional assumptions, such as not allowing triangular inequality violations. This further complicates comparisons of solution methodologies. Therefore in this paper we focus on a general purpose commercial IP solver to obtain exact solutions as they do not require additional assumptions.

We denote by performance measures the measures of a problem that provide insight into the difficulty of solving a problem. Therefore, problem size is by definition the essential performance measure for the worst case analysis of algorithms for combinatorial problems. However, we are interested in understanding how the problem formulation impacts solution times. Hence, we are interested in data based performance measures not related to problem size. We also focus on the observed solution run times and not on the worst case complexity of the algorithm. There exist research on performance measures for NP-hard optimization

problems that aim to explain the observed variation in solution times. It is shown that these problems tend to exhibit sharp changes in computational complexity, with instances going from being easy to being hard to solve (or vice-versa) as a certain performance measure changes. These sharp frontiers are referred to as phase transitions. In particular the work on the traveling salesman problem (TSP) (Gent and Walsh 1996; Cheeseman et al. 1991; Zhang 2004) is related to performance measures for the VRP due to the similarities between the problems. Phase transitions for the TSP have been identified based on asymptotic properties of the optimal tour length as the problem size increases and the existence of solution backbones, i.e. variables that are present in all optimal solutions (Gent and Walsh 1996; Zhang 2004). Our approach is closer to Cheeseman et al. (1991) and Zhang and Korf (1996), which identify phase transitions that depend on properties of the distance matrix of the TSP, such as the range and standard deviation and not on asymptotic properties of the problem.

3 VRP Formulations

Due to the wide range of applications that the VRP has addressed, there are a number of different variants of the VRP, obtained by adding specific features or constraints to the basic capacitated vehicle routing problem (CVRP). Some of the most important variants are: VRP with multiple depots (MDVRP), VRP with time windows (VRPTW), VRP with back-hauls (VRPB), and VRP with pick-up and deliveries (VRPPD). To get a handle on the possible number of explanatory factors for the observed solution performance we limit our study to the simplest CVRP problem.

The CVRP problem considers the problem of routing at minimum cost a uniform fleet of K vehicles, each with capacity C , to service geographically dispersed customers, each

with a deterministic unit demand that must be serviced by a single vehicle. Let V be the set of n demand nodes and a single depot, denoted as node 0. Let d_i be the demand at each node i . Since we assume unit demand $d_i = 1$ for all $i \in V \setminus \{0\}$. We consider the fully connected network, and denote the deterministic travel time between node i and node j by c_{ij} . We assume asymmetric deterministic travel times (i.e. $c_{ij} \neq c_{ji}$ in general) which might in addition violate the triangular inequality.

We now describe different formulations for the CVRP. Our presentation here also follows the review in the book Toth and Vigo (2002b). In broad terms, formulations for the CVRP can be separated into two main categories: path-based formulations and arc-based formulations. Path-based formulations, also referred to as set-partitioning formulations, consider a binary variable for each path (or vehicle route) which indicates whether that route is part of the routing solution or not. These formulations therefore consider a number of integer variables that grow exponentially with n and are typically tackled via column generation schemes. Our work does not consider path-based formulations because of its dependence on specialized algorithms. We present now different arc-based formulations, which we group into polynomial vehicle flow models, commodity flow models, and exponential vehicle flow models.

3.1 Polynomial vehicle flow models

Arc-based models consider integer variables x_{ij} which indicate whether a vehicle goes from node i to node j or not. Problem (1) below includes additional continuous variables u_i for every $i \in V \setminus \{0\}$ that represent the flow in the vehicle after it visits customer i .

$$\text{(VRP1) } \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (1.1)$$

$$\text{s.t. } \sum_{i \in V} x_{ij} = 1 \quad j \in V \setminus \{0\} \quad (1.2)$$

$$\sum_{j \in V} x_{ij} = 1 \quad i \in V \setminus \{0\} \quad (1.3)$$

$$\sum_{i \in V} x_{i0} = K \quad (1.4) \quad (1)$$

$$\sum_{j \in V} x_{0j} = K \quad (1.5)$$

$$u_i - u_j + Cx_{ij} \leq C - d_j \quad i, j \in V \setminus \{0\}, i \neq j, d_i + d_j \leq C \quad (1.6)$$

$$d_i \leq u_i \leq C \quad i \in V \setminus \{0\} \quad (1.7)$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V \quad (1.8) .$$

Problem VRP1 minimizes the total travel time while ensuring a feasible route. The first two constraints, (1.2) and (1.3), force that a single vehicle goes into and out of every node; constraints (1.4) and (1.5) ensure that a total of K vehicles leave the depot and then return to it; constraints (1.6) and (1.7) are subtour elimination constraints imposing both the capacity and connectivity of the feasible routes. Note that isolated subtours violate the subtour elimination constraints and that in our case these constraints simplify slightly since the demand at each node is $d_i = 1$. Although this formulation is polynomial in size, it is well known that it can lead to weak LP relaxations. In fact, Desrochers and Laporte (1991) propose tightening constraints for VRP1.

Our second vehicle flow model considered is known as the three-index formulation. This formulation is derived from VRP1 by adding an index k to the integer variables x_{ijk} to identify which vehicle k traverses arc (i, j) . In addition this formulation considers an integer variable y_{ik} which indicates if vehicle k services customer i or not. Hence the

three-index formulation we consider is

$$(VRP2) \quad \min \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k=1}^K x_{ijk} \quad (2.1)$$

$$\text{s.t.} \quad \sum_{k=1}^K y_{ik} = 1 \quad i \in V \setminus \{0\} \quad (2.2)$$

$$\sum_{k=1}^K y_{0k} = K \quad (2.3)$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} \quad i \in V, k \in 1 \dots K \quad (2.4)$$

$$\sum_{i \in V} d_i y_{ik} \leq C \quad k \in 1 \dots K \quad (2.5) \quad (2)$$

$$i, j \in V \setminus \{0\}, i \neq j,$$

$$u_{ik} - u_{jk} + C x_{ijk} \leq C - d_j \quad \text{s.t.} \quad d_i + d_j \leq C \quad (2.6)$$

$$k \in 1 \dots K$$

$$d_i \leq u_{ik} \leq C \quad i \in V \setminus \{0\}, k \in 1 \dots K \quad (2.7)$$

$$x_{ijk} \in \{0, 1\} \quad i, j \in V, k \in 1 \dots K \quad (2.8)$$

$$y_{ik} \in \{0, 1\} \quad i, \in V, k \in 1 \dots K \quad (2.9) .$$

Constraints (2.2) and (2.3) ensure that exactly one vehicle services every customer $i \in V \setminus \{0\}$ and that K vehicles leave the depot. Constraints (2.4) force vehicle k to arrive and leave from node i only if it services that node. Constraints (2.5) enforce the capacity constraint for every vehicle k , while constraints (2.6) and (2.7) are the subtour elimination constraints.

3.2 Commodity flow models

Commodity flow models were recently introduced for the symmetric CVRP in the paper by Baldacci et al. (2004). Below we describe an asymmetric version of this model. Commodity flow models consider, in addition to the routing variables x_{ij} , two sets of continuous non-negative flow variables y_{ij} and z_{ij} for each arc (i, j) . If a vehicle travels from i to j in the

network, it carries a load y_{ij} and a residual capacity z_{ji} so that $y_{ij} + z_{ji} = C$. The model also requires a dummy depot node $n + 1$ to the network. Let $V' = V \cup \{n + 1\}$ and also let D be the set of demand nodes, so $D \cup \{0, n + 1\} = V'$. The vehicles in the model now depart node 0 and finish at $n + 1$.

$$\text{(VRP3) } \min \sum_{i \in V'} \sum_{j \in V'} c_{ij} x_{ij} \quad (3.1)$$

$$\text{s.t. } \sum_{j \in V'} (y_{ji} - y_{ij}) = d_i \quad i \in D \quad (3.2)$$

$$\sum_{j \in V'} (z_{ji} - z_{ij}) = d_i \quad i \in D \quad (3.3)$$

$$\sum_{j \in D} y_{0j} = d(D) \quad (3.4)$$

$$\sum_{j \in D} z_{j0} = KC - d(D) \quad (3.5)$$

$$\sum_{j \in D} z_{n+1j} = KC \quad (3.6)$$

$$y_{ij} + z_{ji} = C x_{ij} \quad i, j \in V' \quad (3.7)$$

$$\sum_{j \in V'} (x_{ij} + x_{ji}) = 2 \quad i \in D \quad (3.8)$$

$$y_{ij} \geq 0 \quad i, j \in V' \quad (3.9)$$

$$z_{ij} \geq 0 \quad i, j \in V' \quad (3.10)$$

$$x_{ij} \in \{0, 1\} \quad i, j \in V' \quad (3.11)$$

(3)

Constraints (3.2) and (3.3) enforce that the flow and residual capacity variables are adjusted according to the demand at every node $i \in D$; constraints (3.4) send the total demand out of the depot; constraints (3.5) and (3.6) respectively enforce total amounts of residual capacity into the depot and out of the dummy node; constraints (3.7) ensure the flow and residual capacity add up to C for every arc that is selected, and set these variables to zero on arcs that are not used; finally constraints (3.8) are the routing constraints enforcing that exactly a vehicle exactly arrives and departs from every node i .

3.3 Exponential vehicle flow models

Different formulations of the VRP are obtained by replacing the subtour elimination constraints (1.6) and (1.7) in problem (1) with either the capacity-cut constraints (1.9) or the generalized subtour elimination constraints (1.10) below:

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq \gamma(S) \quad S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (1.9) ,$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - \gamma(S) \quad S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (1.10) .$$

Constraint (1.9) enforces that the number of vehicles entering any subset S must be enough to satisfy the demand at S . Constraint (1.10) implies that the number of vehicles that leave S is enough to satisfy the demand at S . Here the function $\gamma(\cdot)$ is defined for every $S \subseteq V$, such that $\gamma(S)$ equals the optimal value for the bin packing problem of demand $d(S) = \sum_{i \in S} d_i$ ($= |S|$) with bins of size C . The function $\gamma(S)$ can be equivalently replaced by the bin packing problem lower bound given by $\lceil d(S)/C \rceil$. In fact, in the case of unit demand this lower bound is tight.

Due to the exponential explosion in the number of constraints, the above models become significantly large as the problem size increases. Therefore in practice, they are not solved completely, but they are used in a constraint generation procedure. Since we are interested in this study on the performance of formulations on a general IP solver, we do not present the results for these exponential models. However, it is known that these models are mostly insensitive to problem parameters since they result in tight LP relaxations.

4 Experimental Results

In this experimental section we first identify, for each formulation, the problem size for which an exact solution can be obtained in a reasonable amount of time to conduct a parameter sensitivity analysis. We then explore the effect of vehicle capacity C and number of vehicles K on problem solution time. We investigate how different aspects of the travel time matrix c_{ij} affect the problem solution time. Our last computational experiment explores the effect of geographical distribution of customers on the solution time.

For each problem formulation, and each setting of problem parameters (demand nodes n , vehicle capacity C , number of vehicles K , and mean μ_c and standard deviation σ_c of the travel time matrix c_{ij}) we create random instances by randomly generating the $n(n + 1)$ off-diagonal values of the travel time matrix following a lognormal distribution with mean μ_c and standard deviation σ_c . The lognormal distribution assumption for actual travel time distances is standard in the transportation literature, see Dessouky et al. (1999). Our computational results considers a base case scenario with the parameter values $C = 6$, $K = 5$, $\mu_c = 4.5$, and $\sigma_c = 3.375$. For each scenario, 50 instances are randomly generated and here we define the problem solution time as the average solution time of these instances.

All experiments were carried out on a Dell Dimension 8200 computer with 2.4 GHz Pentium 4 Processor and 1GB RDRAM running Red Hat Linux 7.3. Exact solutions for the five VRP formulations considered were obtained using the mixed integer programming solver from CPLEX 8.1 with default settings.

4.1 Problem Size

Our goal in selecting a problem size is to obtain runtimes that allow a study of the sensitivity of the solution time with respect to changes in problem parameters, therefore we select different problem sizes for each formulation. After some preliminary computations we observed that VRP2, the three-index formulation, is more difficult to solve than the other polynomial formulations since VRP2 has more than K times the number of integer variables than either VRP1 or VRP3. In the remainder of the paper we use $n = 26$ for VRP1 and VRP3, and $n = 20$ for VRP2. With these sizes, the mean solution time with base case parameters are between 1 to 2 minutes for all formulations. These solution times are small enough so that the experiments can be executed in a reasonable time and also significant enough so that changes to solution time can be noticed.

4.2 Effect of C and K on Solution Time

We plot in the left side of Figure 1 the mean solution time for each of the 3 formulations as we vary the vehicle capacity C . Note that in these experiments, μ_c and σ_c are kept at their base case values of 4.5 and 3.375, respectively and we fix the number of vehicles $K = 5$. We note that as the vehicle capacity increases all three formulations become easier to solve essentially because the problem is becoming less constrained and optimal solutions are simpler. In particular, if the vehicle capacity increases, then it is more likely that efficient routing solutions will have enough capacity to satisfy the demand. We obtained very similar results when we fix the vehicle capacity C and vary the number of vehicles K because as the number of vehicles K increases, each vehicle can perform shorter routes in the optimal solution. We omit these results here for space considerations.

In the right side of Figure 1, we also plot information about the distribution of the 50

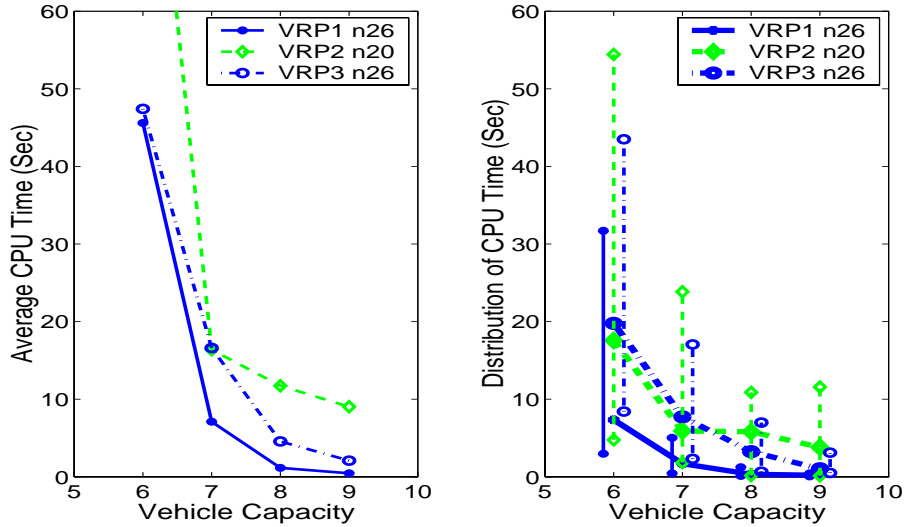


Figure 1: LEFT: Average CPU time as a function of the vehicle capacity. RIGHT: Median, 1st and 3rd quartiles. With $\mu_c = 4.5$, $\sigma_c = 3.375$, $K = 5$, 50 random instances

instances for each data point. For all three formulations, around each median value, we plot vertical lines between the first and the third quartiles of 50 instances. Despite some fluctuations, we believe the averages are good enough representations of the underlying distribution of the 50 instances. We observed similar trends in all of the experimental results and therefore we only present the averages in the reminder of this paper to make the figures clearer.

4.3 Effect of the Travel Time Matrix on Solution Time

To investigate the effect of the travel time matrix on solution time we present in Figures 2, 3, and 4 the mean solution times as we respectively vary only the mean μ_c , standard deviation σ_c , and coefficient of variation $CV_c = \sigma_c/\mu_c$, maintaining $C = 6$ and $K = 5$. Since the coefficient of variation is defined in terms of both μ_c and σ_c , there are many

combinations that yield the range explored in the experiment. In our settings we consider:

μ_c	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5
σ_c	0.625	1.75	3.375	5.5	9.75	15	25.5	38	52.5	69
CV_c	0.25	0.5	0.75	1	1.5	2	3	4	5	6

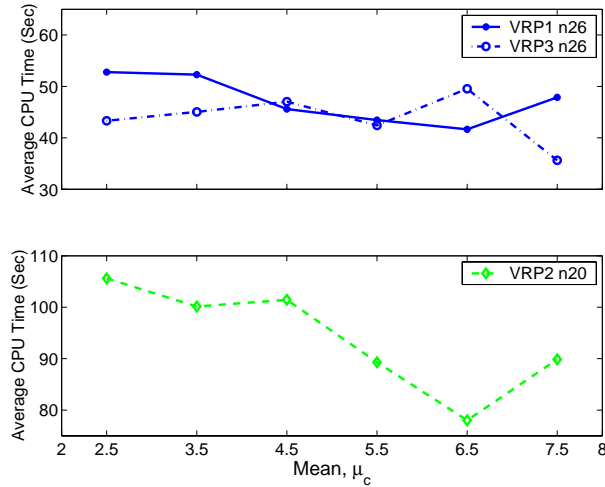


Figure 2: Average CPU time as a function of mean travel time μ_c . With $C = 6$, $K = 5$, $\sigma_c = 3.375$, 50 random instances

For VRP1 and VRP3 the mean does not seem to impact solution times. However, there is a large dependency on σ_c , which shows an increase of more than 25 seconds and then a decrease. The results with respect to CV_c are less clear, as it seems to show a strong increase for small coefficients and to be almost indifferent to larger coefficients. Formulation VRP2 exhibits a different behavior. It decreases significantly with an increase in μ_c and it increases significantly with an increase in σ_c . However for CV_c , the effect is mixed.

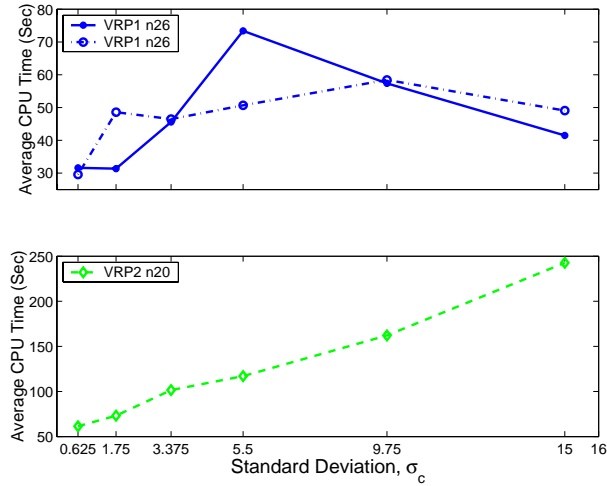


Figure 3: Average CPU time as a function of standard deviation of travel time σ_c . With $C = 6$, $K = 5$, $\mu_c = 4.5$, 50 random instances

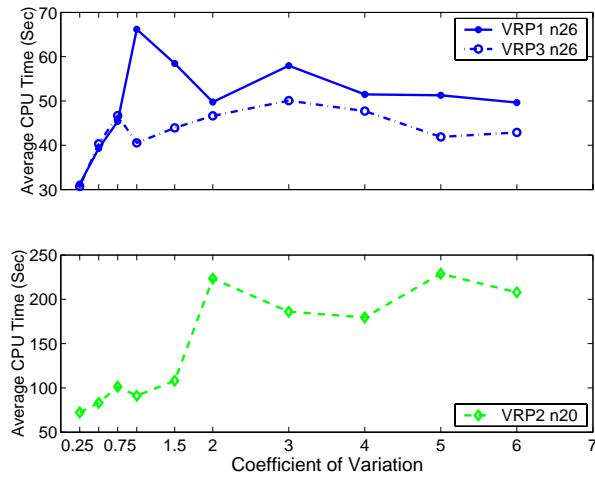


Figure 4: Average CPU time as a function of the coefficient of variation of travel time σ_c/μ_c . With $C = 6$, $K = 5$, 50 random instances

4.4 Understanding the Effect of CV_c on Solution Time

Taking a closer look at the results with respect to varying CV_c in Figure 4, we can explain the split behavior in VRP1, VRP2, and VRP3 (first increase followed by relative independence from CV_c) by the interplay of two competing forces: an increase in the number of triangular inequalities violated and an increase in the asymmetry of the travel time matrix. Before discussing these two competing forces in detail, we comment on how the random problems are generated for the experiments.

For each random instance we generate $n(n+1)$ random lognormal numbers with mean μ_c and standard deviation σ_c . This instance is used for all three formulations. However since each formulation considers a different number of nodes, each formulation uses as many of the generated random numbers as appropriate. We generate $26 * 27$ random lognormal numbers, all of which are used for VRP1 and VRP3, while VRP2 uses the first $20 * 21$ numbers.

The fact that the number of triangular inequalities violated and the asymmetry of the travel time matrix increase as CV_c increases is not surprising. For instance, for a fixed mean, CV_c increases by increasing the standard deviation. If the travel time matrix is generated with a larger variability then it is reasonable to have more triangular inequality violations and a larger distance from symmetric instances. We also show this relationship experimentally in Figure 5. The number of triangular inequalities violated is obtained by checking all inequalities for every subset of three arcs in the matrix. We define the asymmetry of a square matrix A by $\frac{1}{\sqrt{2}}\|A - A^T\|_F = \sqrt{\sum_{i>j}(A_{ij} - A_{ji})^2}$.

We note that the number of triangular inequalities violated increases sharply for small values of CV_c and slower for large CV_c , while the rate of increase of the asymmetry is very linear. Hence for small CV_c the number of triangular inequalities violations will be

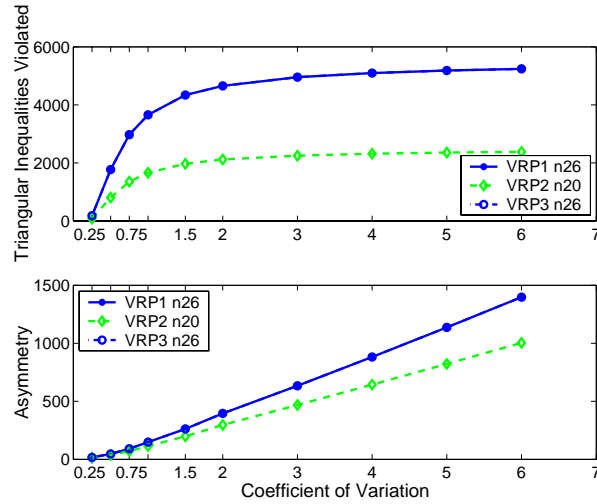


Figure 5: Number of triangular inequalities violated and Asymmetry as a function of the coefficient of variation CV_c . With $C = 6$, $K = 5$, 50 random instances

the dominating factor due to its sharp increase, whereas for greater values of CV_c the asymmetry of the matrix becomes more important. Note also that since VRP1 and VRP3 use the same data, both the number of triangular inequalities violated and the asymmetry are the same for those formulations.

The fact that the number of violated triangular inequalities is positively correlated with solution times is suggested from the existence of better worst case complexity results for the TSP when the distances satisfy triangular inequalities (Christofides 1976; Orponen and Mannila 1987). In particular, we observe experimentally that the number of triangular inequality violations is positively correlated with the initial LP gap, which is of course related to the solution time. Both these trends appear in Figure 6 for VRP1 and VRP3, and also hold for the other formulation VRP2. Note that these plots consider ranges for the values on the x-axis, as it is extremely difficult to construct at random examples with a specific number of violated triangular inequalities or with a given initial LP gap. Therefore,

in these plots, the y values are the mean of a certain number of problems that fall within each range, or bucket. The actual number of problems used varies for each formulation since it is determined by the bucket with fewest problems. For each problem formulation we indicate the number of problems used to compute the mean in the graph legend, next to the formulation name and number of demand points.

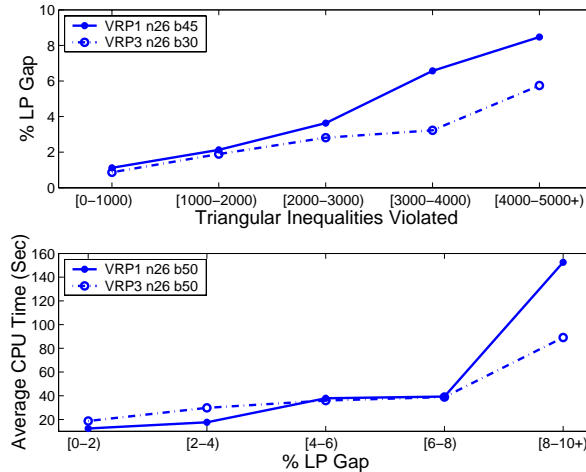


Figure 6: Initial LP gap as a function of triangular inequalities violated and solution time as a function of initial LP gap for VRP1 and VRP3. With $C = 6$, $K = 5$

In conclusion, for small CV_c , the main effect of increasing CV_c is an increase in the number of violated triangular inequalities, which implies an increase in solution time. A phenomenon that is observed in Figure 4 for small CV_c .

For large CV_c we expect the dominating factor to be the asymmetry of the travel time matrix. We observe the effect of this factor on solution time in Figure 7, and note that for VRP1 it becomes easier to solve as asymmetry increases and for VRP2 and VRP3 the effect is mixed showing an increase and then a decrease.

Finally, to validate the fact that there are two competing factors influencing solution

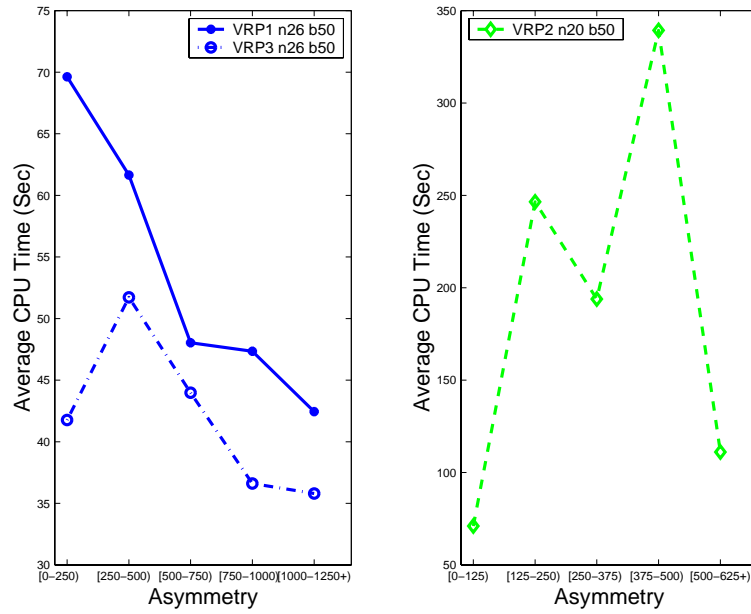


Figure 7: Average CPU time as a function of the asymmetry in the matrix of travel times. With $C = 6$, $K = 5$

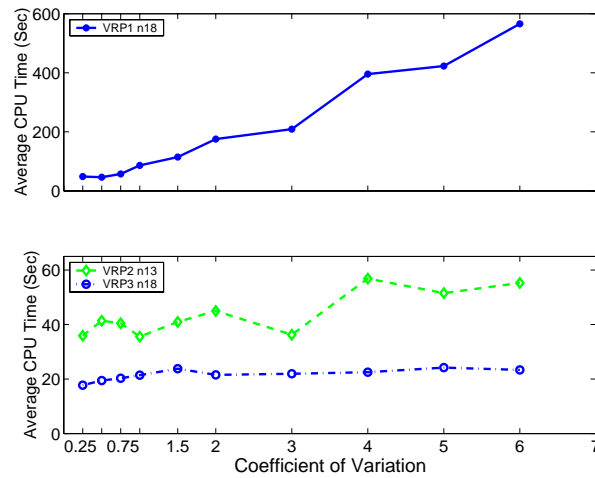


Figure 8: Average CPU time as a function of the coefficient of variation CV_c for symmetric instances. With $C = 6$, $K = 5$, 50 random instances

time as we increase CV_c , we plot in Figure 8 how the solution times are affected by an increase in CV_c for symmetric data instances. We note that for formulations VRP1, and to some extent VRP2, the solution times increase as the coefficient of variation increases. Therefore the fact that the increase stops for large CV_c in Figure 4 is most likely due to the asymmetry. A second observation from Figure 8 is that instances with symmetric data are harder to solve, forcing a reduction in the size of problems considered for formulations VRP2 (from 20 to 13), and VRP1 and VRP3 (from 26 to 18). We note that these symmetric instances were solved using the asymmetric formulations presented earlier, hence the difficulty stems from having many routes with similar total costs, since now traversing a route clockwise or counterclockwise are equivalent.

4.5 Effect of Geographical Distribution of Customers

Lastly, we investigate the effect of geographical distribution of customers on the solution time. For this, we randomly generate customers lying on a 2-dimensional Euclidean plane and then construct the cost matrix. Problems generated with this procedure will have symmetric cost matrix which also obeys the triangular inequalities. We randomly generate customers on the plane with different degrees of clustering. For that, we consider a depot at location $(0, 0)$ and randomly generate customer locations in 5 different clusters (one for each vehicle) with identical radius of $r = 20$. We center each cluster at a random location a distance R from the depot. By increasing the value of R , we change the geographical distribution of customers from uniformly distributed to clustered. We measure how much the customers are clustered with the following clustering ratio: $r_c = \frac{R}{r}$. Figure 9 shows averages of 50 random instances for each clustering ratio. Clearly, VRP3 benefits the most from the effect of clustering while VRP1 benefits less. For those two formulations clustering helps decomposing the problem naturally and makes it easier to solve since each vehicle is

assigned to a single cluster. On the other hand, this has a negative effect on VRP2. Since VRP2 distinguishes between the vehicles, there are several different assignments of vehicles to the clusters resulting in the same total cost and not being able to prune those solutions increases the solution time of VRP2.

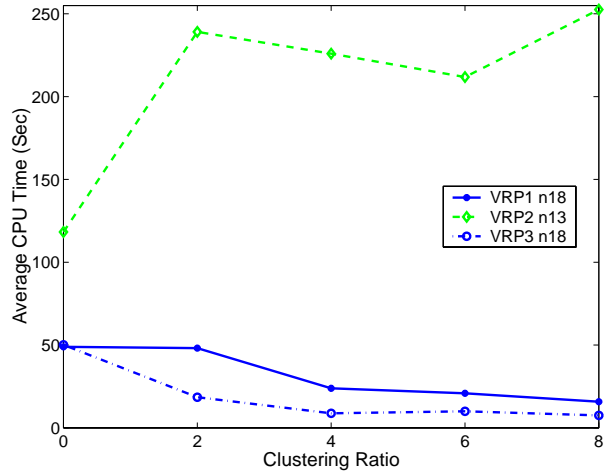


Figure 9: Average CPU time as a function of the clustering ratio r_c for symmetric instances. With $C = 6$, $K = 5$, 50 random instances

5 Conclusions

In this work we investigate the effect of problem formulation and data parameters on the solution run time that a state-of-the-art general purpose IP solver takes to solve the capacitated VRP. The VRP is a key part of operations for an important number of industries, such as courier services, trucking companies, and demand responsive transportation services. As such, it has received substantial attention from the research community, which has generated a number of equivalent formulations of the problem and solution methods.

How to formulate the routing problem at hand then becomes the first question to answer for a practitioner. Being able to identify the formulation that is easier to solve can help develop an overall better solution method because we can focus on algorithms to address the most promising formulations. By concentrating on the performance of a general IP solver we aim to represent an intrinsic difficulty of the problem and in the process identify the best problems for a solution method that has become more relevant both for exact and heuristic solutions.

We find that due to the initial integrality gap of polynomial arc-based formulations make the performance sensitive to the problem parameters. We now summarize our findings of the experiments presented.

- All models show a significant decrease in solution time if we increase the capacity or the number of vehicles.
- The effect of the travel time matrix on solution time is more obscure. VRP2 shows a strong dependency on the standard deviation, while VRP1 and VRP3 show first an increase in solution time, then oscillatory behavior as CV_c increases.
- This mixed dependency on CV_c is explained by two factors. As CV_c increases we have that the number of violated triangular inequalities increases and the asymmetry of the matrix increases. The increase in triangular inequalities violations is most significant for smaller values of CV_c and leads to an increase in solution time, while the asymmetry increase becomes the dominating factor for large CV_c .
- Problem VRP1 becomes easier to solve as the asymmetry increases, while problems VRP2 and VRP3, first show an increase then a decrease in solutions times with asymmetry.

- For symmetric problems, an increase in coefficient of variation leads to an increase in solution times.
- For symmetric problems, when the degree of clustering of the geographical distribution of customers increases, VRP1 and VRP3 become easier to solve whereas VRP2 shows an increase in solution times.

We observed similar performance trends when considering symmetric problems, and have no reason to suspect a deviation from this behavior for the symmetric versions of the formulations considered. However, specialized algorithms that favor a specific formulation, can lead to different results. The question of which formulation has a specialized algorithm that performs best is at the heart of a number of VRP challenges and competitions, focuses on specific problem instances and is not investigated here. For instance, using a path-based formulation is a competitive solution approach for problems with hard time windows, which help reduce the number of feasible solutions. How the problem parameters influence such a path-based formulation still has not been fully addressed.

References

- Agarwal, Y., K. Mathur, and H. M. Salkin (1989). A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks* 19, 731–749.
- Baldacci, R., E. Hadjiconstantinou, and A. Mingozzi (2004). An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research* 52(5), 723–738.
- Cheeseman, F., B. Kanefsky, and W. M. Taylor (1991). Where the really hard problems are. In *Proceedings NCAI-91*, Sydney, pp. 331–337.

- Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh.
- Christofides, N. and S. Eilon (1969). An algorithm for the vehicle dispatching problem. *Operational Research Quarterly* 20, 309–318.
- Christofides, N., A. Mingozzi, and P. Toth (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming* 20(3), 255–282.
- Clarke, G. and J. Wright (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12(4), 568–581.
- Cordeau, J. F., M. Gendreau, G. Laporte, J. Y. Potvin, and F. Semet (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53(5), 512–522.
- Desrochers, M., J. Desrosiers, and M. M. Solomon (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* 40, 342–354.
- Desrochers, M. and G. Laporte (1991). Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters* 10, 27–36.
- Dessouky, M., R. Hall, A. Nowroozi, and K. Mourikas (1999). Bus dispatching at timed transfer transit stations using bus tracking technology. *Transportation Research Part C* 7, 187–208.
- Dorronsoro Díaz, B. (2005). The VRP web. <http://neo.lcc.uma.es/radi-aeb/WebVRP>.
- Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum k -trees. *Operations Research* 42, 626–642.

- Gendreau, M., A. Hertz, and G. Laporte (1994). A tabu search heuristic for the vehicle routing problem. *Management Science* 40, 1276–1290.
- Gent, I. P. and T. Walsh (1996). The TSP phase transition. *Artificial Intelligence* 88, 349–358.
- Gillett, G. and L. Miller (1974). A heuristic algorithm for the vehicle dispatch problem. *Operations Research* 22, 340–349.
- Hadjiconstantinou, E., N. Christofides, and A. Mingozzi (1995). A new exact algorithm for the vehicle routing problem based on q -paths and k -shortest paths relaxations. *Annals of Operations Research* 61, 21–43.
- Homberger, J. and H. Gehring (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR* 37(3), 297–318.
- Jones, K. L., I. J. Lustig, J. M. Farwolden, and W. B. Powell (1993). Multicommodity network flows: The impact of formulation on decomposition. *Mathematical Programming* 62, 95–117.
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59, 345–358.
- Laporte, G. and Y. Norbert (1987). Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics* 31, 147–184.
- Lenstra, J. K. and A. H. G. Rinnooy Kan (1981). Complexity of vehicle routing and scheduling problems. *Networks* 11, 221–227.
- Lu, Q. and M. Dessouky (2004). An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science* 38(4), 503–514.
- Martinhon, C., A. Lucena, and N. Maculan (2004). Stronger k -tree relaxations for the vehicle routing problem. *European Journal of Operational Research* 158, 56–71.

- Orponen, P. and H. Mannila (1987). On approximation preserving reductions: Complete problems and robust measures. Technical Report C-1987-28, Department of Computer Science, University of Helsinki.
- Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41, 421–451.
- Palmer, K., M. M. Dessouky, and T. Abdelmaguid (2004). Impacts of management practices and advanced technologies on demand responsive transit systems. *Transportation Research, Part A: Policy and Practice* 38, 495–509.
- Potvin, J. Y. and S. Bengio (1996). The vehicle routing problem with time windows. part ii: Genetic search. *INFORMS Journal on Computing* 8(2), 165–172.
- Ralphs, T. (2003). Parallel branch and cut for vehicle routing. *Parallel Computing* 29(5), 607–629.
- Ralphs, T., L. Kopman, W. Pulleyblank, and L. Trotter (2003). On the capacitated vehicle routing problem. *Mathematical Programming* 94(2-3), 343–359.
- Toth, P. and D. Vigo (2002a). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics* 123, 487–512.
- Toth, P. and D. Vigo (Eds.) (2002b). *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications. SIAM.
- Zhang, W. (2004). Phase transitions and backbones of the asymmetric traveling salesman problem. *Journal of Artificial Intelligence Research* 21, 471–497.
- Zhang, W. and R. E. Korf (1996). A study of complexity transitions on the asymmetric traveling salesman problem. *Artif. Intell.* 81(1-2), 223–239.