VRP Bench: A Vehicle Routing Benchmark Tool

October 19, 2016

Guilherme A. Zeni¹, Mauro Menzori¹, P. S. Martins¹, Luis A. A. Meira¹

Abstract

The number of optimization techniques in the combinatorial domain is large and diversified. Nevertheless, there is still a lack of real benchmarks to validate optimization algorithms. In this work we introduce VRP Bench, a tool to create instances and visualize solutions to the Vehicle Routing Problem (VRP) in a planar graph embedded in the Euclidean 2D space. We use VRP Bench to model a real-world mail delivery case of the city of Artur Nogueira. Such scenarios were characterized as a multi-objective optimization of the VRP. We extracted a weighted graph from a digital map of the city to create a challenging benchmark for the VRP. Each instance models one generic day of mail delivery with hundreds to thousands of delivery points, thus allowing both the comparison and validation of optimization algorithms for routing problems.

Keywords: Benchmark, VRP, Graphs and Combinatorial Optimization, Free Software Tool.

¹School of Technology, University of Campinas, Limeira – SP – Brazil
1 Introduction

Benchmarks are found in various fields of science, such as geology [13], economy [25], climatology [36], among other areas. Specifically in computer science, benchmarks play a central role, e.g. in image processing [15, 29, 22], hardware performance [7] and optimization [32, 28, 5].

In the context of optimization, David S. Johnson [24] divided algorithm analysis in three approaches: the worst-case, the average-case, and the experimental analysis. Relative to experimental papers, he identifies four situations: (i) to solve a real problem; (ii) to provide evidence that an algorithm is superior than the others; (iii) to better understand a problem; and (iv) to study the average-case. He suggests the use of well-established benchmarks to provide evidence of the superiority of an algorithm (item ii). Such papers are called horse race papers.

Johnson highlights that reproducibility and comparability are essential aspects present in any experimental paper. He also advocates the use of instances that lead to general conclusions. The author mentions the difficulty in justifying experiments on problems with no direct application. Such problems have no real instances and the researcher is forced to generate the data in a vacuum. Johnson cautions against a pitfall: the researcher starts by using randomly-generated instances to evaluate the algorithms and ends up using the algorithm to explore the properties of the randomly-generated instances. According to him, another pitfall is spending time processing useless experiments that attempt to answer the wrong questions.

Our work deals with a variant of the Vehicle Routing Problem (VRP) based on a real mail delivery case of the city of Artur Nogueira (see the contributions at the end of this section). One of the first references to the VRP dates back to 1959 [14] under the name Truck Dispatching Problem, a generalization of the Traveler Salesman Problem (TSP).

The term VRP was first seen in the paper by Christophides [9], in 1976.
Christophides defines VRP as a generic name, given to a class of problems that involves the visit of “customers” using vehicles.

Real world aspects may impose variants of the problem. For example, the Capacitated-VRP (CVRP) \[18\] considers a limit to the vehicle capacity, the VRP with Time Windows (VRPTW) \[26\] accounts for delivery time windows, and the Multi-Depot VRP (MDVRP) \[34\] extends the number of depots. Other variants may be easily found in the literature.

In 1991, Reinelt \[32\] created a benchmark for the TSP, known as TSPLib. In his work, he consolidated non-solved instances from twenty distinct papers. His repository (TSPLIB95) \[33\] has instances of both the symmetric and the asymmetric traveling salesman problem (TSP/aTSP) as well as three related problems: (i) CVRP; (ii) Sequential Ordered Problem (SOP); and (iii) Hamiltonian Cycle Problem (HCP).

The number of instances is 113, 19, 16, 41, 9 for TSP, aTSP, CVRP, SOP, and HCP, respectively. The number of vertices varies from 14 to 85900 for the TSP, 17 to 443 for the aTSP, 7 to 262 for the CVRP, 7 to 378 for the SOP, and from 1000 to 5000 for the HCP.

The optimum of all TSPLib instances was finally achieved in 2007, after sixteen years of notable progress in the development of algorithms. The optimum of the d15112 instance was found in 2001 \[2\]. This instance contains 15112 German cities and required 22.6 years of processing split across 110 500MHz processors \[12\]. The instance pla33810 was solved \[2\] in March 2004. The pla33810 instance represents a printed circuit board with 33810 nodes and it was solved in 15.7 years of processing \[16\]. The last instance of the TSPLib, called pla85900, was solved in 2006 \[2\]. This instance contains 85900 nodes representing a VLSI application.

Solomon \[35\] created a benchmark for the VRPTW in 1987. It is composed by 56 instances partitioned in 6 sets. The number of customers is 100 in all instances. The vehicle has a fixed capacity and the customers
have demand or weight. The number of vehicles is not fixed, but it derives the fact that there is a limit of capacity. In this view, the problem can be considered multi-objective. It aims to minimize the route and the number of vehicles.

The first optimum solution was published in 1999 [27]. In 2005, Chabrier [6] managed to solve 17 of the instances that still had remained unsolved in the benchmark. In 2010, Amini et. al [1] obtained solutions very close to optimum, considering the first 25 customers only. In july 2015, after 28 years of the benchmark launching, Jawarneh and Abdullah [23] published in PlosONE a Bee Colony Optimization metaheuristic. Such algorithm reached 11 new best results in Solomon’s 56 VRPTW 100 customer instances. It is really surprising that such small instances have an internal structure so complex to be optimized. In Figure 1, there is a simple instance of Solomon’s composed by 100 customers and its solution considering 3 vehicles.

![Figure 1: The Solomon’s RC207 instance composed by 100 customers and 1 depot. Bent and Hentenryck proposed the solution containing 3 routes, in 2001 [4]. The picture does not show the capacity and the time windows constraints.](image)

Despite of their complexity, the TSPLib and the Solomon benchmarks
have a number of customers between 100 and 262 for the VRP, which is a small value nowadays. Gehring and Homberger [19] extended the instances of Solomon, creating a benchmark with a numbers of customers varying from 100 to 1000 for the VRPTW.

For the CVRP, there is a set of instances largely used, named as ABEFMP, in which Augerat [3] proposed the classes A, B, P in 1995 and [10, 17, 8] proposed the classes E, F, M in 1969, 1994 and 1979, respectively. In this benchmark, the number of customers varies from 13 to 200 and the number of vehicles varies from 2 to 17.

A series of different works, such as [18, 11] among others, obtained the optimum in different ABEFMP instances. [31] found in 2014 the optimum solution for the last instance unsolved, named $M-n151-k12$, 35 years after its presentation by Christofides [8]. Despite that, most of those instances are very simple to solve nowadays.

Golden et al. [20] proposed new instances for the CVRP, in 1998. It is a set of 20 instances, with the number of customers varying from 240 to 483. Such a benchmark remains entertaining, because most of its instances has not an optimum established yet [38]. In 2005, Li, Golden and Wasil [30] created a set of instances with the number of customers between 560 and 1200. Up to this moment, there is no optimum defined for any of the instances [38].

In 2014, Uchoa et. al. created a library, the CVRPLib [38]. In this library, they consolidated the CVRP instances of the works [3, 10, 8, 17, 20, 30]. Besides, Uchoa et. al [37] generated new instances with the number of customers between 100 and 1000. In this work, they point the lack of well established challenging benchmarks for the VRP.

Uchoa et. al. also point the fact that benchmarks are created artificially. Solomon and Uchoa et. al. generated their own instances using random points. In the benchmark ABEFMP, there is some random instances generated and other instances that represent real problems. However, in all
instances the customers are points in the Euclidean space. The instances of Golden et al. [20] and Li, Golden and Wasil [30] are artificial as well.

Contribution: In this work, we propose the VRPBench tool, a vehicle routing problem benchmark tool for the mail delivery Problem. We propose instances for the VRP involving a real situation: the delivery of correspondence by the postmen in the Artur Nogueira road network. Artur Nogueira is a Brazilian city, located at 22°34′22″S 47°10′22″W. Every day, the post office gets a large number (thousands) of letters to be delivered. The letters are distributed to a set of 20-25 postmen. The postmen walk around the city, delivering the mail.

The problem described above is modeled as VRP. Each postman is a vehicle and each delivery point is a customer. We abstracted the streets and avenues of the city in an weighted undirected graph. The street corners and the delivery points both became vertices and the streets became edges. The weight of an edge is proportional to the time needed to travel it.

After talking to experts, two objectives were detected to minimize: the average route length; and the injustice, measured as the unbalancing weight between the postmen.

The postmen do not have capacity, but the routes do have. Each postman can carry 10kg if man or 8kg if woman, however there is a car support that supplies the load to the postmen, which turns their capacities unlimited in mathematical terms. Meanwhile, each postman must follow a working day close to 6h per day, which generates a maximum capacity for the route.

Defining capacity in the route length and not in the vehicle is something that applies to several situations. A helicopter has a route limited by its fuel tank. The workers, in general, have a time window to operate the vehicle, which limits the length of the route.

According to our knowledge, this is one of the first benchmarks for the VRP with thousands of customers and that models a real-life situation in a
road network.
2 Notation and Definition

Consider a weighted (directed or undirected) graph $G(V,E)$ where a depot is a special vertice $\pi \in V$. This research do not address multiple deposits variants of VRP. Consider a cost function $w' : E \to \mathbb{Q}^+$

The set of customers is defined by $C = V \setminus \{\pi\}$ and the number of customers is denoted by $n$, where $C = \{c_1, \ldots, c_n\}$. The number of vehicles in the fleet is represented by $k \in \mathbb{N}$. The value $k$ is traditionally considered a constant, but it is possible to define variants of VRP where $k$ is variable.

Let $w(u,v,G)$ be the shortest path cost between vertices $u$ and $v$ in the graph $G$. We use $w(u,v)$ to represent $w(u,v,G)$ whenever $G$ is known within the context. We represent a solution as a sequence the vertices. Let

$$S(C,k) = (c_1, \ldots, c_n, \pi, \ldots, \pi).$$

This sequence is created as follows: (i) all the customers are inserted in $S$; (ii) the deposit vertice is inserted $k - 1$ times.

Each permutation of $S(C,k)$ represents a solution to the VRP. For example, consider the graph illustrated in Fig. 2:

![Figure 2: Sample VRP Vertices.](image)

Consider the vertices described in Fig. 2 and suppose that the number
of vehicles is 3 (i.e. \( k = 3 \)). The \( S(C, 3) \) sequence is:

\[
S(C, 3) = (c_1, \ldots, c_{13}, \pi, \pi).
\]

Each permutation of \( S(C, 3) \) represents a solution to the VRP. For example,

\[
S' = (c_3, c_5, c_4, c_1, c_2, \pi, c_6, c_{10}, c_{11}, c_{12}, \pi, c_7, c_8, c_9, c_{13})
\]

is the solution described in Fig. 3.

Figure 3: Solution \( S' = (c_3, c_5, c_4, c_1, c_2, \pi, c_6, c_{10}, c_{11}, c_{12}, \pi, c_7, c_8, c_9, c_{13}) \).

All routes begin and end at the depot. The \( S' \) solution represents a partition of the clients in 3 routes: \( \mathcal{R}_1 = (c_3, c_5, c_4, c_1, c_2) \), \( \mathcal{R}_2 = (c_6, c_{10}, c_{11}, c_{12}) \) and \( \mathcal{R}_3 = (c_7, c_8, c_9, c_{13}) \). The vertex \( \pi \) is used to create a partition of the sequence in \( k \) routes:

\[
\text{Partition}(S) = (\mathcal{R}_1, \ldots, \mathcal{R}_k)
\]

where the sequence is split in each \( k - 1 \) occurrence of \( \pi \).

In the example shown in Fig. 3, we have

\[
\text{Partition}(S') = ((c_3, c_5, c_4, c_1, c_2), (c_6, c_{10}, c_{11}, c_{12}), (c_7, c_8, c_9, c_{13}))
\]
The length of a route $R = (r_1, \ldots, r_m)$ is given by:

$$W(R) = w(\pi, r_1) + w(r_m, \pi) + \sum_{i=1}^{m-1} w(r_i, r_{i+1}).$$

The length of a solution $S = (s_1, \ldots, s_m)$ is calculated in as

$$W(S) = w(\pi, s_1) + w(s_m, \pi) + \sum_{i=1}^{m-1} w(s_i, s_{i+1}).$$

Given a viable solution, it is necessary to calculate its cost $f(S)$. The most traditional objective function to be minimized is the length of the solution:

$$f(S) = W(S).$$

However, in some situations the problem is considered as multi objective. One of the objectives is to minimize $W(S)$ and the other is to minimize the number of vehicles.

$$f_1(S) = W(S), \quad f_2(S) = k.$$

Assume a system that generates mailman routes. It is required that the solution meets the fairness criteria, i.e., routes are assigned in a way that are not unfair to the mailman, i.e. it does not penalize one mailman in relation to the other. One way to achieve fairness would be to minimize the variance of the length of the routes for each mailman.

As described in Section 1, VRP is a set of problems that consist in visiting customers using vehicles. Clearly, there are several variants of the VRP problem. In each, the cost of the solution is calculated differently. Furthermore, each case has additional requirements to find out whether or not a solution is viable.

Examples of restrictions to the problem are:
There should be no empty routes, i.e. for each route $\mathcal{R}_i \in \text{Partition}(\mathcal{S})$ we must meet the condition that $|\mathcal{R}_i| > 0$.

Each customer $c \in C$ has a demand $d(c)$ (e.g. number of orders). Each vehicle in $v \in \{1, \ldots, k\}$ is characterized by a capacity $c(v)$. This variant of VRP models situations where a driver (e.g. mailman) has a maximum weight limit. Therefore, a route $\mathcal{R}_v = (r^v_1, \ldots, r^v_m)$ must meet the requirement that

$$\sum_{i=1}^{m} d(r^v_i) \leq c(v), \forall v \in \{1, \ldots, k\}. $$

Consider a variant of the VRP problem where the size of the route is limited. This problem can model the scenario where vehicles need to fuel at the depot (e.g. a helicopter). It also cater for legal labour issues, where a driver has a commitment to minimum availability. Let $v \in \{1, \ldots, k\}$ be a vehicle with a maximum route $\mathcal{R}_{\max}(v)$. Therefore, any given route $\mathcal{R}_v = (r_1, \ldots, r_m)$ is constrained by the condition that

$$W(\mathcal{R}_v) \leq \mathcal{R}_{\max}(v), \forall v \in \{1, \ldots, k\}. $$

VRPTW is a variant of the VRP problem, where a visit to a vertex must be carried out within a time window. In this case, the solution is deemed to be feasible if a client is within a pre-defined lower and upper time limit. Examples of cases are scheduled deliveries such as SEDEX 10, which guarantees that the delivery is completed by the following day before 10 am. It is also useful to implement residential technical support, where the visit to a customer is scheduled within a time frame.

A simple way to deal with unfeasible solutions is to assign an arbitrary high cost to the objective functions: whenever a solution is considered unfeasible, its cost is infinite ($f(\mathcal{S}) \leftarrow \infty$).

For simplicity, in Section 3 we will describe the incapacitated VRP with time windows. The number of vehicles is a constant and we seek to minimize
the total length of the routes ($W(S)$). We also assume there are no empty
or idle routes.

**Definição 1 (Single-objective VRP)** Given a weighted graph $G(V,E)$,
a constant $k$, a special vertex $\pi \in V$ and an objective function $f$. Seja
$C \leftarrow V \setminus \{\pi\}$. Consider the sequence $S(C,k)$ and let $P$ be the set of all
permutations of $S(C,k)$. Find the permutation $S^* \in P$ so that $f(S^*)$ is
minimum.
3 Methodology

The instances were not generated from the actual post-office application data. Nevertheless, the first author has applied his domain expertise (having worked for four years in the field at a post office in Arthur Nogueira) to turn the instances as realistic as possible. The starting point was the map of the city of Arthur Nogueira as shown in Fig. 4.

Figure 4: A Section of the Map of Artur Nogueira.

Line segments representing the streets were drawn on top of the map (Fig. 5). Each corner was automatically identified by means of an algorithm that calculate intersections. The result was a weighted graph where the weight of an edge represents the length of the segment.

Clearly, the edges representing deliveries that are carried out by a vehicle need to follow the direction of one-way streets. However, in our case, postmen deliveries were performed by foot.

We have created the graph directly from the street map due to the fact that: (i) the path by foot may differ from the ones available from maps which prioritize delivery by vehicles; (ii) the number of streets in Artur Nogueira is slightly over 400 and that allows the manual creation of the graph; (iii) Currently, public maps such as OpenStreetMap [21] are incomplete, i.e.
they have a large number of streets not yet covered;

Ideally, the creation of new benchmarks for other cities should also consider public maps, mainly in large cities where the manual extraction of the map may incur in prohibitive costs.

The final graph ended up with $|V| = 2111$ and $|E| = 3225$. Each edge is associated with a pixel. Each edge has a bijection with a line segment. Each street has a name and a set of collinear line segments. The cost of an edge is proportional to the length of its associated line segment, as discussed later.

An initial approach would be to consider the deliveries generated according to a uniform distribution across the city streets. However, a non-uniform distribution would be more realistic. For example, the main streets in the downtown area do tend to receive a larger number of deliveries per unit length than the side or back streets.

Therefore, a relative density $D$ parameter was applied to each street. For example, a street with $D = .5$ has a 50% less probability of receiving a delivery than one with $D = 1.0$ (per unit length).

### 3.1 Streets Density of Probability

For each street, we arbitrarily classify it with attributes and levels.
There are three attributes Region (R), Type (T) and Zone (Z). Both attributes and penalties were created based on expert knowledge.

The criterion by regions follows the premise that the downtown area’s closest streets have a higher rate of deliveries by length unit than the streets located at extreme regions of the city.

We define a multiplicative penalty for each type of region.

| Region      | central | peripheral | distant | isolated |
|-------------|---------|------------|---------|----------|
| Multiplic. Penalty (Penal) | 1.0 | 0.75 | 0.4 | 0.2 |

It means that, fixing other attributes, an isolated street receives 80% less deliveries than a central street per unit of length.

We also define the attribute Type with four levels:

| Type     | Avenue | Street | Alameda | Highway |
|----------|--------|--------|---------|---------|
| Multiplic. Penalty (Penal) | 1.0 | 0.75 | 0.4 | 0.0 |

In this work, all highways received directly the value 0, because there were no deliveries on it.

A street has 25% menos entregas por unidade de comprimento do que uma avenida.

At last, we created the attribute Zone, which has three levels: with penalties

| Zone         | Commercial&Industrial | Mixed | Residential |
|--------------|-----------------------|-------|-------------|
| Multiplic. Penalty (Penal) | 1.0 | 0.7 | 0.4 |

We used the Google Maps as an auxiliary tool to verify the streets containing companies.

The criterion Zone exists because a street deliveries density does not depend only on its type or region. The economic context should also be regarded.

Each one of the 422 streets received a value in $Rg \times T \times Z$, according to expert knowledge. Be $Streets = (Street_1, Street_2, \ldots, Street_{422})$ the set of
all streets. Be \( f : Streets \rightarrow Rg \times T \times Z \) a function that attributes values to a particular street. For instance, consider the \( Street_{RuaXV} \). We have \( f(Street_{RuaXV}) \) was manually define as \( (central, Avenue, Mixed) \), because it is an avenue, spotted at downtown, containing residences and commercial establishments.

We can refer to each \( f \) element apart, in the following manner: \( Rg(Street_{XV}) = central, T(Street_{XV}) = Avenue \) and \( Z(Street_{XV}) = Mixed. \)

Finally, we have a function that describes the relative density of a street. 
\[ D : Streets \rightarrow \mathbb{N}, \text{as it follows:} \]
\[ D(Street) = Penal(Rg(Street)) \times Penal(T(Street)) + Penal(Z(Street)). \]

Using this methodology, all the streets received a relative density. For example, the XV de Novembro Avenue relative density is given by
\[ D(Street_{XV}) = Penal(Rg(Street_{XV})) + Penal(T(Street_{XV})) + Penal(Z(Street_{XV})) \]
\[ = Penal(central) \times Penal(Avenue) \times Penal(Mixed) \]
\[ = 1 \times 1 \times 0.7 = 0.7 \]

(1)

On the other hand, the Sibipirunas Alameda has the relative density:
\[ D(Street_{Sib.}) = Penal(Rg(Street_{Sib.})) + Penal(T(Street_{Sib.})) + Penal(Z(Street_{Sib.})) \]
\[ = Penal(isolated) + Penal(Alameda) + Penal(Residential) \]
\[ = 0.2 \times 0.2 \times 0.4 = 0.16 \]

(2)

\( D(Street_{XV}) = 0.7 \) and \( D(Street_{Sib.}) = 0.16 \) are used so that, in our model, the probability density of the \( Rua_{XV} \) are be \( \frac{0.7}{0.14} \) times larger than probability density \( Street_{Sib.} \).
3.2 Generating the Delivery Points

We shall define a variable weight $w$ as

$$w(\text{Street}) = D(\text{Street}) \times \text{Length}(\text{Street}).$$

So $w$ is directly proportional to the relative probability density and to the street length.

$$E[ne(\text{Street})] = w(\text{Street}) k,$$

Take a sequence of weights $(w_1, w_2, \ldots, w_n)$ associated to the streets $(\text{Street}_1, \text{Street}_2, \ldots, \text{Street}_n)$.

Thus, we will define a variable $W$ as

$$W = \sum_{i=1}^{n} w(\text{Street}_i)$$

The algorithm considers all the weights projected in a range $[0, W]$. For each delivery point, a random value $R \in [0, W]$ is created. Case this random value falls on the range of a particular street, the delivery point will be inserted in the respective street.

In a street, the $ne(\text{Street})$ delivery points are uniformly distributed.

4 Using VRP Bench to Model Manhattan Streets

5 Result

We generated 10 sets of instances with 10 instances for set, resulting 100 instances. The delivery points number of each set is 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000 and 10000, respectively.

The figure below shows the distribution of 1000 delivery points, for a given area of the city chosen randomly:
**Input**: A sequence of weights \((w_1, w_2, \ldots, w_n) \in \mathbb{R}^n\) associated to streets and the total number of delivery \(m \in \mathbb{N}\).

**Output**: The number of delivery for each street
\((ne_1, ne_2, \ldots, ne_n) \in \mathbb{N}^n\)

1. Set \(W = \sum_{i=1}^{n} w(\text{Street}_i)\)
2. for \(i = 1\) to \(m\) do
3. \(R \leftarrow \text{random value int [0,W]}\)
4. for \(x = 1\) to \(m\) do
5. \(\text{if } \sum_{j=1}^{x-1} w_j < R \leq \sum_{j=1}^{x} w_j \text{ then}
\)
6. \(ne_x \leftarrow ne_x + 1\)
7. end
8. end
9. end
10. return \((ne_1, ne_2, \ldots, ne_n)\)

**Algorithm 1**: Generating delivery points.

### 6 Conclusions

In this work, we created a benchmark based on a real-life situation. Once the solution used by post office is fixed with respect to the delivery points, there is a large gap to be explored.

A feasible solution that reduces the number of postmen produces profit, since such postmen can be allocated in other tasks.

A solution that reduces the route length also reduce the delivery effort, and also can be understand as profit.

Furthermore, in science view, this is the first benchmark to VRP with 10,000 delivery points.
Future Perspectives

From this work, we aim to publish a paper. We intend to develop a free software system to validate and to rank the best solutions found by the researchers, also providing a view of the routes. Lower bounds have a more complex validation and will be released only when associated to a publishing.

At last, I would like to work in the development of an algorithm to solve some of the instances presented in this work, in a future master degree.

References

[1] Shahrzad Amini, Hassan Javanshir, and Reza Tavakkoli-Moghaddam. A pso approach for solving vrptw with real case study. Int. J. Res. Rev. Appl. Sci, 4(3):118–126, 2010.

[2] David L Applegate, Robert E Bixby, Vasek Chvatal, and William J
Cook. *The Traveling Salesman Problem: A Computational Study: A Computational Study*. Princeton university press, 2011.

[3] Ph Augerat, JM Belenguer, E Benavent, A Corberán, D Naddef, and G Rinaldi. *Computational results with a branch and cut code for the capacitated vehicle routing problem*. IMAG, 1995.

[4] Russell Bent and Pascal Van Hentenryck. A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science*, 38(4):515–530, 2004.

[5] Rainer E Burkard, Stefan E Karisch, and Franz Rendl. Qaplib—a quadratic assignment problem library. *Journal of Global optimization*, 10(4):391–403, 1997.

[6] Alain Chabrier. Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research*, 33(10):2972–2990, 2006.

[7] Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W Sheaffer, Sang-Ha Lee, and Kevin Skadron. Rodinia: A benchmark suite for heterogeneous computing. In *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*, pages 44–54. IEEE, 2009.

[8] N Christofides. The vehicle routing problem. combinatorial optimization. christofides n., mingozzi a., toth p., sandi c.(eds) j, 1979.

[9] Nicos Christofides. The vehicle routing problem. *Revue française d’automatique, d’informatique et de recherche opérationnelle. Recherche opérationnelle*, 10(1):55–70, 1976.

[10] Nicos Christofides and Samuel Eilon. An algorithm for the vehicle-dispatching problem. *Or*, pages 309–318, 1969.
[11] Claudio Contardo and Rafael Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014.

[12] William Cook. Traveling salesman problem site. (last check 01/30/2016) [http://www.math.uwaterloo.ca/tsp/d15sol/](http://www.math.uwaterloo.ca/tsp/d15sol/), 2016.

[13] M Correia, J Hohendorff, ATFS Gaspar, D Schiozer, et al. Unisim-ii-d: Benchmark case proposal based on a carbonate reservoir. In *SPE Latin American and Caribbean Petroleum Engineering Conference*. Society of Petroleum Engineers, 2015.

[14] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[15] JM Hans du Buf, M Kardan, and Michael Spann. Texture feature performance for image segmentation. *Pattern recognition*, 23(3):291–309, 1990.

[16] Daniel G Espinoza. *On linear programming, integer programming and cutting planes*. PhD thesis, Georgia Institute of Technology, 2006.

[17] Marshall L Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations research*, 42(4):626–642, 1994.

[18] Ricardo Fukasawa, Humberto Longo, Jens Lysgaard, Marcus Poggi de Aragão, Marcelo Reis, Eduardo Uchoa, and Renato F Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical programming*, 106(3):491–511, 2006.

[19] Hermann Gehring and Jörg Homberger. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In *Proceedings of EUROGEN99*, volume 2, pages 57–64. Springer Berlin, 1999.
[20] Bruce L Golden, Edward A Wasil, James P Kelly, and I-Ming Chao. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In Fleet management and logistics, pages 33–56. Springer, 1998.

[21] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. IEEE Pervasive Computing, 7(4):12–18, 2008.

[22] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[23] Sana Jawarneh and Salwani Abdullah. Sequential insertion heuristic with adaptive bee colony optimisation algorithm for vehicle routing problem with time windows. PloS one, 10(7):e0130224, 2015.

[24] David S Johnson. A theoretician’s guide to the experimental analysis of algorithms. Data structures, near neighbor searches, and methodology: fifth and sixth DIMACS implementation challenges, 59:215–250, 2002.

[25] Philippe Jorion. Value at risk: the new benchmark for controlling market risk. Irwin Professional Pub., 1997.

[26] Brian Kallehauge, Jesper Larsen, Oli BG Madsen, and Marius M Solomon. Vehicle routing problem with time windows. Springer, 2005.

[27] Niklas Kohl, Jacques Desrosiers, Oli BG Madsen, Marius M Solomon, and Francois Soumis. 2-path cuts for the vehicle routing problem with time windows. Transportation Science, 33(1):101–116, 1999.

[28] Rainer Kolisch and Arno Sprecher. Psplib-a project scheduling problem library: Or software-orsel operations research software exchange program. European journal of operational research, 96(1):205–216, 1997.
[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[30] Feiyue Li, Bruce Golden, and Edward Wasil. Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, 32(5):1165–1179, 2005.

[31] Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. In *Integer programming and combinatorial optimization*, pages 393–403. Springer, 2014.

[32] Gerhard Reinelt. Tsp-lib—a traveling salesman problem library. *ORSA journal on computing*, 3(4):376–384, 1991.

[33] Gerhard Reinelt. Tsp-lib95. *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Heidelberg*, 1995.

[34] Jacques Renaud, Gilbert Laporte, and Fayez F Boctor. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, 23(3):229–235, 1996.

[35] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[36] Richard SJ Tol. Estimates of the damage costs of climate change. part 1: Benchmark estimates. *Environmental and resource Economics*, 21(1):47–73, 2002.

[37] Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Anand Subramanian, and Thibaut Vidal. New benchmark
instances for the capacitated vehicle routing problem. unpublished. http://www.galgos.inf.puc-rio.br/vrp/attachments/article/8/New_Instances.pdf, Last checked 15-Jun-2016.

[38] Eduardo Uchoa, Diego Pecin, Artur Pessoa, Marcus Poggi, Anand Subramanian, and Thibault Vidal. Cvrp library site. http://www.galgos.inf.puc-rio.br/vrp/, Last checked 23–Fev–2016, 2015.