# kVRP (K-customer Vehicle Routing Problem) Survey

Huy Doan Ravindra Manjunatha Chin Wei Yeap
University of Texas at Austin

In this paper, we will survey the common approximation approaches towards K-customer Vehicle Routing problem. We will briefly discuss about many variants of kVRP. Then we will talk about $\frac{1}{2}-$ differential approximation for non-metric kVRP and $min\left\{\frac{2}{3},\frac{k-1}{k+1}\right\}$-differential approximation algorithm for metric kVRP and follow it up with an improved algorithm which provides tighter bound of above approximation.
Keywords: Vehicle Routing Problem (VRP), k-Vehicle Routing Problem (kVRP), differential approximation

## 1 Introduction

Consider a complete undirected simple graph G with $V(G) = 0, 1, \ldots, n$ and edge-weight $d_{i,j}$ for each edge $(i, j) \in E(G)$. We call the vertex 0 depot, and the other vertices customers. A route is a closed walk which is either a simple cycle containing the vertex 0, or a 2-edge walk $(0, i, 0)$ for some $i \neq 0$. The objective is to compute a set of routes $\{C_1, \ldots, C_p\}$ minimizing the total weight such that $|C_i| \leq k+1$ for every $i = 1, \ldots, p$, $V(C_i) \cap V(C_j) = \{0\}$ for every $i, j (i \neq j)$, and $\cup_{i=1}^{p} V(C_i) = V(G)$ [8]. It is a generalized version of Travelling Salesman Problem (TSP). For k$\geq$ 3, metric kVRP and unweighted kVRP were proved NP-hard by Haimovich and Rinnooy Kan [5] and Hassin and Rubunstein [6] respectively. In this survey paper, we will provide an overview of the kVRP, review the differential cost, the ratio between the (upper bound - $\alpha$) and the (lower bound - $\alpha$), and provide proof snippet. We will also draw some insights and conclusions from our observations.

## 2 Background

kVRP and VRP are extensively studied operation research problems. Many algorithms and concepts in kVRP share similarities with the more generalized VRP problem. The traditional approaches focus on solving problem using exact algorithms, which can be classified into 3 broad categories: (i) direct tree search methods; (ii) dynamic programming, and (iii) integer linear programming [7]. We will discuss a few exact algorithms below. Branch-and-bound method limits the depth or hops of each route while brand-and-cut method limits the breadth or branches of each route at intersection. Clarke and Wright (1964) introduced the greedy Savings algorithm to start with any route and then merge the routes starting with the maximum savings until the incremental savings become minimal [3]. After extensive research and refinement, the 2-step divide and conquer approaches were introduced to solve the VRP problem. They were known as cluster-then-route and route-then-cluster methods. Cluster-then-route method first groups all delivery stops in cluster and then uses Minimum Spanning Tree (MST) as route to connect the clusters. On the other hand, route-then-cluster method starts with a Traveling Salesman Problem (TSP) tour and then merges the nearby delivery stops in clusters. Both methods build routes in loops of loops where outer loop is a global TSP tour across all clusters and inner loops are local TSP tour within each cluster. Petal algorithm is also widely used in VRP. It can be 1-petal or 2-petal. 1-petal method has 1 TSP tour on the seed vertex of the cluster. 2-petal has 2 different TSP tours on the seed vertex of the cluster, so it is more flexible and can cover wider area. Gillett and Miller (1974) introduced Sweep algorithm, a cluster-then-route method with capacity constraint [4]. It can be divided to forward sweep and backward sweep methods. Forward sweep method builds cluster starting from first stop then extending to subsequent stops. Backward sweep method builds cluster in reverse order by first visiting all the stops and then builds cluster around them. Nowadays, kVRP and VRP problems are solved with more heuristic and dynamic approaches.

In this survey paper, we will only focus on approximation methods for kVRP. kVRP focuses on serving at most K customers in each tour. It includes two versions: metric and non-metric. Metric version assumes triangle inequality to hold while non-metric version does not follow triangle inequality. There are other variants such as capacitated (CVRP), time-window (TWVRP), distance-constraint (DVRP), online, offline, and very offline kVRP. CVRP, TWVRP, and DVRP have constraints on routing infrastructure cost. CVRP limits each vehicle with capacity contraint. Time-window VRP has the specific delivery window for each customer. DVRP has the distance constraint of each vehicle route. Precedence VRP (PVRP), online kVRP, offline kVRP, and very offline kVRP have constraints on routing delivery order. Precedence VRP specifies the preferred delivery order of each customer. Online kVRP serves the request instantly as soon as the request comes in. Offline kVRP receives all the requests together initially and plans accordingly. The very offline VRP is more flexible and can serve the requests in any order.

# 3 Differential Approximation of kVRP

Beside the standard measurement $\alpha - approximation$ where $\alpha = \frac{approx}{opt}$, $approx$ and $opt$ are the approximation and optimal solutions respectively, another measurement is differential approximation in which $\alpha = \frac{wor-approx}{wor-opt}$, $wor$ is the worst cost or the optimal solution of the complementary problem. In this section, we present some recent research on both non-metric kVRP and metric kVRP problems using differential approximation.

## 3.1 Non-metric kVRP

We will introduce a $\frac{1}{2} - approximation$ algorithm for the non-metric VRP using the knowledge of binary 2-matching. Binary 2-matching is a subgraph in which every vertex has a degree of exactly 2. A minimum binary 2-matching is one with minimum sum of all edge weights [1]. First we generate a graph $G'$ from the original graph $G$ by replacing the depot, node 0, with a complete graph $G_0 = (V_0, E_0)$ where $|V_0| = 2n$ and all edges in $E_0$ have zeo weight. The edge weights between a vertex in $V_0$ and vertex i $\in V \setminus V_0$ is the same as that is between vertex 0 and i. We then can generate a minimum binary 2-matching $M'$ of the graph $G'$ in $O(n^3)$ [5]. It is proved that the weight of $M'$, LB, is a lower bound of the kVRP problem, i.e. $opt \geq LB$ [2]. And the binary 2-matching cycle $M'$ in $G'$ can be transformed to a set of cycles M that cover vertices in G with the same weight. This is straight forward to see. Considering a cycle $C'$ in $M'$, there are three cases:

- $C'$ does not contain any vertices in $V_0$: $C'$ is a cycle in $M$.
- $C'$ contains consecutive vertices in $V_0$: we can combine those vertices into one in $V_0$
- $C'$ has the form $(v_0^1, \mu_1, v_0^2, \ldots, \mu_2, v_0^p, \mu_t, v_0^1)$: $M$ has p cycles $(0, \mu_1, 0), (0, \mu_2, 0, ), \ldots, (0, \mu_p, 0)$ and the total weight of all cycles in $C$ is the same as that of $C'$

Now we introduce the $\frac{1}{2} - approximation$ algorithm for $k \geq 3$ using the previous knowledge about binary 2-matching and the transformation [2].

**Algorithm 1** Differential kVRP

**procedure** DIFFERENTIAL_KVRP($G = (V, E)$)
  Compute $G'$
  Compute minimum binary 2-matching $M'$ of $G'$
  Transform $M'$ to $M = \{C_1, C_2, \ldots, C_p\}$ ▷ $LB \leftarrow d(M)$
  **for all** $C_i = (1, \ldots, m_i, 1) \in M$ **do** ▷ For the first case
    **if** $m_i$ is even **then**
      $sol_{i,1} \leftarrow \{(0,1,2,0),(0,3,4,0),\ldots,(0,m_i-1,m_i,0)\}$
      $sol_{i,2} \leftarrow \{(0,m_i,1,0),(0,2,3,0),\ldots,(0,m_i-2,m_i-1,0)\}$
    **else**
      $sol_{i,1} \leftarrow \{(0,1,2,0),(0,3,4,0),\ldots,(0,m_i-4,m_i-3,0)\} \cup \{0,m_i-2,m_i-1,0\}$
      $sol_{i,2} \leftarrow \{(0,m_i,1,0),(0,2,3,0),\ldots,(0,m_i-2,m_i-1,0)\}$
  **for all** $C_i = (0, 1, \ldots, m_i, 0) \in M$ with $m_i > k$ **do** ▷ For the second and third cases
    **if** $m_i$ is even **then**
      $sol_{i,1} \leftarrow \{(0,2,3,0),\ldots,(0,m_i-2,m_i-1,0)\} \cup \{(0,1,0),(0,m_i,0)\}$
      $sol_{i,2} \leftarrow \{(0,1,2,0),\ldots,(0,m_i-1,m_i,0)\}$
    **else**
      $sol_{i,1} \leftarrow \{(0,2,3,0),\ldots,(0,m_i-1,m_i,0)\} \cup \{(0,1,0)\}$
      $sol_{i,2} \leftarrow \{(0,1,2,0),\ldots,(0,m_i-2,m_i-1,0)\} \cup \{(0,m_i,0)\}$
  **for all** $C_i = (0, 1, \ldots, m_i, 0) \in M$ with $m_i \leq k$ **do**
    $sol_{i,1} \leftarrow sol_{i,2} \leftarrow C_i$
  $apx \leftarrow \cup_{i=1}^{p} argmin(d(sol_{i,1}), d(sol_{i,2}))$ **return** $apx$

**Proof:** Let $add_{i,j}$ be the added weight of the solution $soli, j$ for $j = 1,2$ with respect to the length of $C_i$. Denote a feasible solution $sol_{i,3}$ of $C_i$ when $0 \in C_i$ and $|C_i| \leq k+1$ such that $d(sol_{i,3}) = d(sol_{i,1}) + d(sol_{i,2})$. From the algorithm we have $apx = \sum_i^p C_i + min(add_{i,1}, add_{i,2}) = LB + \delta_1$ and $bad = \sum_i^p C_i + d(sol_{i,3}) = LB + add_{i,1} + add_{i,2} = LB + \delta_2$. Clearly that $\delta_2 \geq 2\delta_1$. In this minimization problem, we observe that $wor \geq bad \geq apx \geq opt \geq LB$. Therefore

$$\alpha = \frac{wor - apx}{wor - opt} \geq \frac{bad - apx}{bad - opt} \geq \frac{\delta_2 - \delta_1}{bad - LB} = \frac{\delta_2 - \delta_1}{\delta_2} = 1 - \frac{\delta_1}{\delta_2} \geq \frac{1}{2} \quad \square \tag{1}$$

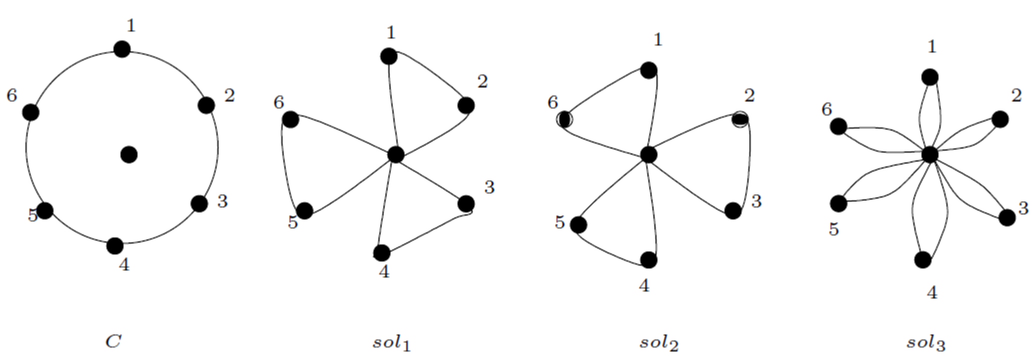


Figure 1: Sets of routes obtained from $C_i$ with $k = 3$ and $m_i = 6$

### 3.2 Metric kVRP

Using the metric that satisfies the triangular inequality, [1] showed that the *Metric kVRP* is $min(\frac{2}{3}, \frac{k-1}{k+1})-$differential approximation [2]. Definition of various symbols are summarised in Table 1.

**Algorithm 2** Differential Metric-kVRP

**procedure** DIFFERENTIAL_METRIC-KVRP($G_1 = (V_1, E_1)$) ▷ $G_1$ is induced by $\{1, \dots, n\}$
  Compute binary 2-matching $M = \{C_1, C_2, \dots, C_p\}$ of $G_1$
  **for all** $C_i = (1, \dots, m_i, 1) \in M$ **do**
    **if** $m_i \leq k$ **then**
      **for** $j = 1$ to $m_i$ **do**
        $sol_{i,j} \leftarrow C_i \setminus \{(j, j+1)(mod\ m_i)\} \cup \{(j,0),(j+1,0)\}$
    **else if** $m_i = kp + r$ $p \geq 1$ and $0 \leq r \leq k-1$ **then**
      **for** $j = 1$ to $m_i$ **do**
        $(\mu_1, \dots, \mu_{m_i\ mod\ k}) \leftarrow C_i \setminus [\{(j-1, j)\} \cup \{(j-1+r+kl, j+r+kl)\}]$ where $1 \leq l \leq p$
    $sol_i \leftarrow argmin\{d(sol_{i,1}, \dots, d(sol_{i,m_i})\}$
  $apx \leftarrow \cup_{i=1}^{p} sol_i$
  **return** $apx$

**Proof:**

- $m_i \leq k$: $apx_i = min_j d(sol_{i,j}) \leq \frac{m_i - 1}{m_i} d(C_i) + \frac{1}{m_i} wor_i \leq \frac{2}{3} d(C_i) + \frac{1}{3} wor_i$.
- $m_i \geq k$ or $m_i = kp + r$:
  - $r = 0 : apx \leq \frac{p(k-1}{kp} d(C_i) + \frac{p}{kp} wor_i \leq \frac{2}{3} d(C_i) + \frac{1}{3} wor_i$
  - $r \geq 1 : apx \leq \frac{p(k-1)+r-1}{kp+r} d(C_i) + \frac{p+1}{kp+r} wor_i \leq \frac{k-1}{k+1} d(C_i) + \frac{2}{k+1} wor_i$

We have

$$\begin{aligned} apx &= \sum_i^p apx_i \\ &\leq \sum_i^p min\left\{\frac{2}{3} d(C_i) + \frac{1}{3} wor_i, \frac{k-1}{k+1} d(C_i) + \frac{2}{k+1} wor_i\right\} \\ &= min\left\{\frac{2}{3} d(M) + \frac{1}{3} wor, \frac{k-1}{k+1} d(M) + \frac{2}{k+1} wor\right\} \end{aligned} \tag{2}$$

Recall that $\alpha = \frac{|wor - apx|}{|wor - opt|}$ and $LB = d(M) = \sum_i^p d(C_i)$, we have:

$$\begin{aligned} \alpha &= min\left\{\frac{\left|\frac{2}{3} d(M) - \frac{2}{3} wor\right|}{|opt - wor|}, \frac{\left|\frac{k-1}{k+1} d(M) - \frac{k-1}{k+1} wor\right|}{|opt - wor|}\right\} \\ &\leq min\left\{\frac{2}{3}, \frac{k-1}{k+1}\right\} \end{aligned} \tag{3}$$

□

### 3.3 Improvement on the Non-metric kVRP

A later improvement by [8] that the non-metric kVRP can have the bound of the metric kVRP without requiring the triangular inequality to hold. Briefly the improve non-metric kVRP algorithm is $min\left(\frac{2}{3}, \frac{k-1}{k+1}\right)$-differential approximation. The new algorithm has a very similar approach to the one mentioned earlier but it constructs two set of solutions *good* and *bad*; $sol_{good,i}$ and $sol_{bad,i}$ for each cycle $C_i$ and $sol_{good} = \cup_{i=1}^{p} sol_{good,i}$ and $sol_{bad} = \cup_{i=1}^{p} sol_{bad,i}$. Denote $\delta_{good,i}$ and $\delta_{bad,i}$ as the added weights of $sol_{good,i}$ and $sol_{bad,i}$ with respect to the weight of $C_i$; similar to the concept of $add_{i,j}$ from the old algorithm. For every edge $e = (i, j) \in C_i$ where $i, j \neq 0$, let $add_e = d_{0,i} + d_{0,j} - d_{i,j} \geq 0$ because $M'$ was computed to have the minimum weight.

**Algorithm 3** Improve kVRP

**procedure** IMPROVE-KVRP($G = (V,E)$)
  Compute $G'$
  Compute minimum binary 2-matching $M'$ of $G'$
  Transform $M'$ to $M = \{C_1, C_2, \ldots, C_p\}$
  **for all** $C_i = (1, \ldots, m_i, 1) \in M$ and $m_i \leq k$ **do** ▷ First case
    **for** $l = 1$ to $m_i$ **do**
      **if** $l < m_i$ **then**
        $sol^l_{good,i} \leftarrow (1, 2, \ldots, l, 0, l+1, \ldots, m_i, 1)$
      **else**
        $sol^l_{good,i} \leftarrow (1, \ldots, m_i, 0, 1)$
    $sol_{good,i} \leftarrow argmin_l\{sol^l_{good,i}\}$
    $sol_{bad,i} \leftarrow \{(0,1,0), (0,2,0), \ldots, (0, m_i, 0)\}$
  **for all** $C_i = (1, \ldots, m_i, 1) \in M$ and $m_i > k$ **do** ▷ Second case
    $C'_i = C_i \setminus \{(m_i, 1)\} \cup \{(m_i, 0), (0, 1)\}$
    **for** $l = 1$ to $k-1$ **do**
      **if** $l < k$ **then**
        $sol^l_{good,i} \leftarrow C_i \setminus \{(j, j+1)\} \cup \{(j, 0), (j+1, 0)\}$ for each $1 \leq j \leq m_i - 1, l \equiv j \bmod k$
      **else**
        $sol^l_{good,i} \leftarrow (1, \ldots, m_i, 0, 1)$
    $sol_{good,i} \leftarrow argmin_l\{sol^l_{good,i}\}$
    $sol_{bad,i} \leftarrow \{(0,1,0), (0,2,0), \ldots, (0, m_i, 0)\}$
  **for all** $C_i = (0, 1, \ldots, m_i, 0) \in M$ and $m_i > k$ **do** ▷ Third case
    **for** $l = 0$ to $m_i$ **do**
      **if** $l < k$ **then**
        $sol^l_{good,i} \leftarrow C_i \setminus \{(j, j+1)\} \cup \{(j, 0), (j+1, 0)\}$ for each $1 \leq j \leq m_i - 1, l \equiv j \bmod k$
      **else**
        $sol^l_{good,i} \leftarrow (1, \ldots, m_i, 0, 1)$
    $sol_{good,i} \leftarrow argmin_l\{sol^l_{good,i}\}$
    $sol_{bad,i} \leftarrow \{(0,1,0), (0,2,0), \ldots, (0, m_i, 0)\}$
  $sol_{good} \leftarrow \cup_{i=1}^{p} sol_{good,i}$
  $sol_{bad} \leftarrow \cup_{i=1}^{p} sol_{bad,i}$
  **return** $sol_{good}$

| Symbol | Definition | Symbol | Definition |
|---|---|---|---|
| M, M’ | binary 2-matching set | $C_i$ | cycle |
| 0, 1, ..., n | vertices | $\delta_i$ | total added weight to the solution |
| $d()$ | length function | $d_{i,j}$ | length of path from i to j |
| $\mu_i$ | a path | $opt, wor$ | values of the optimal and worst solutions |

Table 1: Table of definitions

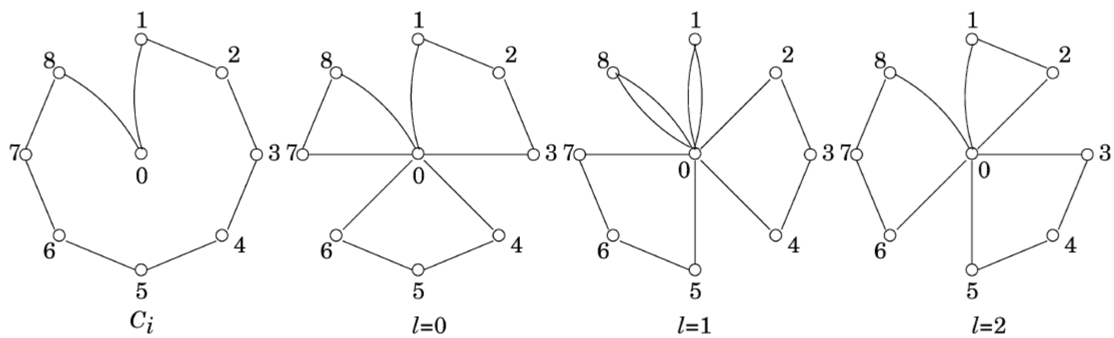


Figure 2: Sets of routes obtained from $C_i$ with $k = 3$ and $m_i = 8$

**Proof:** Denote $e_j = (j, j+1)$ and $e_{m_i} = (m_i, 1)$.

- First case: $\delta_{good,i} \leq \frac{1}{m_i} \sum_{e \in C_i} add_e$ and $\delta_{bad,i} = \sum_{e \in C_i} add_e$.
  Therefore $\delta_{good,i} \leq \frac{1}{3} \delta_{bad,i} \leq max \left\{ \frac{1}{3}, \frac{2}{k+1} \right\} \delta_{bad,i}$ since $k \geq 3$.
- Second case: $\delta_{good,i} \leq \frac{1}{k} \left( \sum_{j=1}^{m_i - 1} add_{e_j} + kadd_{e_{m_i}} \right) = \left( \sum_{j=1}^{m_i} add_{e_j} + (k-1) add_{e_{m_i}} \right)$. And $\delta_{bad,i} = \sum_{j=1}^{m_i} add_{e_{m_i}}$.
  Therefore $\delta_{good,i} \leq \left( \sum_{j=1}^{m_i} add_{e_j} + (k-1) add_{e_{m_i}} \right) = \frac{1}{k} (\delta_{bad,i} + (k-1) add_{e_{m_i}}) \leq \frac{1}{k} \left( \delta_{bad,i} + (k-1) \frac{\delta_{bad,i}}{m_i} \right) = \frac{1}{k} \left( 1 + \frac{k-1}{m_i} \right) \delta_{bad,i}$.
  Since $m_i > k$, $\delta_{good,i} \leq \frac{1}{k} \left( 1 + \frac{k-1}{k+1} \right) \delta_{bad,i} \leq \frac{2}{k+1} \delta_{bad,i} \leq max \left\{ \frac{1}{3}, \frac{2}{k+1} \right\} \delta_{bad,i}$
- Third case: $\delta_{good,i} \leq \frac{1}{k} \sum_{j=1}^{m_i - 1} add_{e_j}$ and $\delta_{bad,i} = \sum_{j=1}^{m_i - 1} add_{e_j}$.
  Therefore $\delta_{good,i} \leq \frac{1}{k} \delta_{bad,i} \leq max \left\{ \frac{1}{3}, \frac{2}{k+1} \right\} \delta_{bad,i}$.

Our final solutions are $sol_{good} \leftarrow \cup_{i=1}^{p} sol_{good,i}$ and $sol_{bad} \leftarrow \cup_{i=1}^{p} sol_{bad,i}$. Therefore $\delta_{good} \leq max \left\{ \frac{1}{3}, \frac{2}{k+1} \right\} \delta_{bad}$. And $\alpha = 1 - \frac{\delta_{good}}{\delta bad} = min \left\{ \frac{2}{3}, \frac{k-1}{k+1} \right\}$ □

# 4 Conclusion

In this paper, we show a brief survey on the kVRP problem and the progress on differential approximation algorithms focusing on solving both metric and non-metric versions of the problem when $k \geq 3$. We present the first algorithm for the non-metric version and prove that it is $\frac{1}{2}-$differential approximation. Similarly we present the algorithm and proof for the metric-version with the performance guarantee of

$min\left\{\frac{2}{3}, \frac{k-1}{k+1}\right\}$. Lastly, we show that is is possible to achieve the bound of the metric version on the non-metric one without requiring the triangular inequality by applying some changes to the orginal algorithm.

## References


[1] Cristina Bazgan, Refael Hassin & Jérôme Monnot (2003): *Differential Approximation for Some Routing Problems*, pp. 277–288. Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/3-540-44849-7_31. Available at `http://dx.doi.org/10.1007/3-540-44849-7_31`.

[2] Cristina Bazgan, Refael Hassin & Jrme Monnot (2005): *Approximation algorithms for some vehicle routing problems*. Discrete Applied Mathematics 146(1), pp. 27 – 42, doi:https://doi.org/10.1016/j.dam.2004.07.003. Available at `http://www.sciencedirect.com/science/article/pii/S0166218X04002860`.

[3] G. Clarke & J. W. Wright (1964): *Scheduling of Vehicle Routing Problem from a Central Depot to a Number of Delivery Points*. Operations research: INFORMS 12, pp. 568–581, doi:10.1287/opre.12.4.568. Available at `http://dx.doi.org/10.1287/opre.12.4.568`.

[4] B. Gillett & L. Miller (1974): *A Heuristic for the Vehicle Dispatching Problem*. Operations Research 22, pp. 340–349, doi:10.1287/opre.22.2.340. Available at `http://pubsonline.informs.org/doi/abs/10.1287/opre.22.2.340`.

[5] M. Haimovich & A. H. G. Rinnooy Kan (1985): *Bounds and Heuristics for Capacitated Routing Problems*. Mathematics of Operations Research 10(4), pp. 527–542, doi:10.1287/moor.10.4.527. Available at `http://dx.doi.org/10.1287/moor.10.4.527`.

[6] Refael Hassin & Shlomi Rubinstein (2005): *On the complexity of the k-customer vehicle routing problem*. Operations Research Letters 33(1), pp. 71 – 76, doi:https://doi.org/10.1016/j.orl.2004.04.003. Available at `http://www.sciencedirect.com/science/article/pii/S0167637704000513`.

[7] Gilbert Laporte (1992): *The vehicle routing problem: An overview of exact and approximate algorithms*. European Journal of Operational Research 59(3), pp. 345 – 358, doi:http://dx.doi.org/10.1016/0377-2217(92)90192-C. Available at `http://www.sciencedirect.com/science/article/pii/037722179290192C`.

[8] Takayuki Nagoya (2009): *New Differential Approximation Algorithm for K-customer Vehicle Routing Problem*. Inf. Process. Lett. 109(8), pp. 405–408, doi:10.1016/j.ipl.2008.12.018. Available at `http://dx.doi.org/10.1016/j.ipl.2008.12.018`.