

有容量限制的车辆路径问题启发式模型的比较研究

Agung Chandra*, Aulia Naro

默库布阿纳大学工程学院工业工程系 雅加达 印度尼西亚

【摘要】 CVRP 是 VRP 的变体，可用于查找最小距离和车辆数量。本文对三种初始解算法进行了比较，以找出货物从配送中心到西雅加达所有常规门店的最短距离—改进的 Clarke 和 Wright (ICW)算法，Karagul Tokat Aydemir (KTA)算法，以及扫描-聚类优先路线第二算法。结果表明，与其他两种算法相比，扫描算法的总距离最短，比 KTA 算法短 48.57%，比 ICW 算法短 33.33%。需要评估更大的样本量来加强这一发现。

【关键词】 CVRP；ICW；KTA；扫描算法

1 引言

车辆路径问题(VRP)及其变体在配送管理或货物运输领域有着非常重要的贡献。许多公司都面临着与人员、货物或信息运输相关的问题。运输成本约占总物流成本的三分之一或三分之二，这就是为什么提高效率是一个主要问题^[1]。无论配送的类型是什么，都必须产生最小的成本^[2]。换句话说，就是在激烈的竞争中实现成本效率和成本效益^[3]。VRP 的一个变体是有容量限制的车辆路径问题—CVRP。

CVRP 是车辆路径问题 VRP 最常见的变体^[4]，是最初的 VRP 问题的基本改进^[5]。CVRP 有两种变体，一种是同质变体—称为 CVRP 统一舰队，另一种是异质变体—也称为 CVRP 混合舰队。在同质变体中，每辆车都有相同的容量，在异质变体中，每辆车都有自己的容量^[6]。CVRP 被归类为 NP(不确定多项式)一困难问题。CVRP 的基本概念是寻找最小化行驶距离和车辆总数的路径^[7]。路径是车辆必须访问的一系列位置或访问的客户，以及它所提供的服务的指示。车辆必须在车库开始和结束其行程^[8,9]。CVRP 有一个额外的约束，如车辆的容量约束和不同节点的可变需求。服务提供商必须在正确的地点向正确的人提供服务，并在正确的时间提供正确数量的物资，以赢得客户的满意^[10]。路线必须满足每个顾客必须被访问一次的约束，顾客的需求被完全满足，并且不得超过每条路线的车辆容量。

在本案例研究中，将对 CVRP 的三种启发式方法进行比较研究：改进的 Clarke 和 Wright (ICW)算法、Karagul Tokat Aydemir (KTA)算法和扫描算法-聚类优先路线第二算法。这三种方法对 VRP 的初始解都有好处。ICW 是 Clarke 和 Wright 提出的一种改进方法—CW 算法。CW 算法被广泛应用于求解 CVRP，自 1964 年提出以来，一直被使用^[11]。

KTA 算法是一种基于物理的优化算法，用于获得 VRP 的初始解，初始解与最佳已知解的平均偏差约为 30%^[12,13]。

扫描算法—聚类优先路线第二算法有一个优点，可以在合理的时间内产生好的解^[14]。扫描算法是聚类优先路线第二的很好的例子^[15]。

CVRP 的这三种算法将在本案例研究中进行比较和应用，本案例研究在该公司定期持续进行。研究将集中在从配送中心到西雅加达所有门店的配送。

2 方法

本研究将比较 CVRP 的三种启发式算法。

2.1 一种改进的 Clarke 和 Wright 算法—ICW 算法

这个方法是由 T. Pichpibul 和 R. Kawtummachai 提出的。该算法用 84 个问题实例进行了测试，其中 68 个实例找到了最优解，其解与最优解之间的平均偏差很低，仅为 0.14%。

ICW 是一种迭代改进方法，旨在找到全局最优解。只有当目前的解决方案优于先前的解决方案时，新的节约列表才会替换先前的节约列表。样本大小 T 是 2 到 6 之间的一个随机数。从节约列表中选择一组节约值，并通过轮盘选择过程挑选出来。节约数 n 、节约值 s_n 、选择概率 p_n 和累积概率 q_n 之间的关系为：

$$p_n = \frac{s_n}{\sum_{i \in T} s_i}, n \in T$$

$$q_n = \sum_{i \in n} p_i, n \in T$$

这个过程被称为两阶段选择过程。流程图如下图 1 所示：

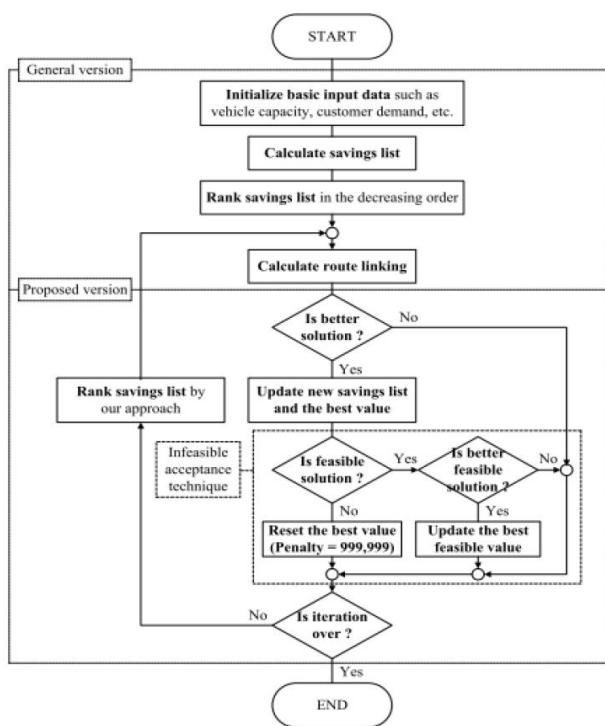


图1 改进的 Clarke 和 Wright 算法

选择的方法是从用一个随机数 r 旋转轮盘开始, r 的范围在零和圆锥之间。如果 $r \leq q_1$, 选择第一个节约值 s_1 ; 而如果 $r \geq q_1$, 选择第 n 个节约值, s_n ($2 \leq n \leq t$), 例如 $q_{n-1} \leq r \leq q_n$ 。所选节约值列在新节约列表中。并且为了避免重复, 从下一个样本选择操作中删除所选择的值。重复这个过程直到最后的节约值从节约列表里挑选出来。

一组客户 v_i 和 v_j 的节约值 s_{ij} 计算如下:

$$s_{ij} = c_{oi} + c_{oj} - c_{ij}$$

在本文中, 这种节约被定义为当两个客户使用同一辆车服务时所节省的距离。

2.2 Karagul Tokat Aydemir—KTA 算法

这个算法起源于创始人的名字: Karagul Tokat dan Aydemir。它的基础是牛顿的质量引力定律。该算法被归类为人工物理优化算法。该方法基于以下等式:

$$X_i^c = \frac{q_i \cdot d_i}{\sum q_i} \text{ 其中 } i = 2, 3, 4, \dots, n$$

X_i = 仓库的质量重力, 其中 i 是客户坐标

q_i = 客户需求的数量($i = 1$ 商店定义)

d = 客户仓库的距离(距离矩阵的第一行和 $n-1$ 个客户编号)

检查仓库和客户位置之间的上述等式状态, 并进行力计算。

$$X_{ij} = \frac{q_i \cdot d_i + q_j \cdot d_j}{(q_i + q_j) \cdot d_{ij}} \text{ 其中 } i = 2, 3, 4, \dots, n-1; j = i+1, \dots, n$$

上述等式显示了客户所在位置之间的关系, 同时考虑了仓库和客户地点之间的质量强度^[16]。

表1 权重矩阵的创建

需求(q_i)		q_1	q_2	q_3	q_4	q_5	q_6
客户	商店	C1	C2	C3	C4	C5	C6
1 商店	-	X_2^c	X_3^c	X_4^c	X_5^c	X_6^c	X_7^c
2 C1		-	X_{23}	X_{24}	X_{25}	X_{26}	X_{27}
3 C2			-	X_{34}	X_{35}	X_{36}	X_{37}
4 C3				-	X_{45}	X_{46}	X_{47}
5 C4					-	X_{56}	X_{57}
6 C5						-	X_{67}
7 C6							-

表2 质量重力矩阵解

需求(q_i)		1	2	3	4	5	6
客户	商店	C1	C2	C3	C4	C5	C6
1 商店	-	X_2^c	X_3^c	X_4^c	X_5^c	X_6^c	X_7^c
2 C1	X_2^c	X_2^c	X_{23}	X_{24}	X_{25}	X_{26}	X_{27}
3 C2	X_3^c	X_{23}	X_3^c	X_{34}	X_{35}	X_{36}	X_{37}
4 C3	X_4^c	X_{24}	X_{34}	X_4^c	X_{45}	X_{46}	X_{47}
5 C4	X_5^c	X_{25}	X_{35}	X_{45}	X_5^c	X_{56}	X_{57}
6 C5	X_6^c	X_{26}	X_{36}	X_{46}	X_{56}	X_6^c	X_{67}
7 C6	X_7^c	X_{27}	X_{37}	X_{47}	X_{57}	X_{67}	X_7^c

这个算法有两个阶段^[17,18]:

(1) 准备阶段:

a. 计算车库-顶点质量力

b. 计算质量引力常数

c. 计算顶点-顶点质量

d. 创建质量力矩阵

e. 将车库-顶点质量力值分配给质量力矩阵中的对角线

单元

(2) 实施阶段:

a.选择第一行中的最小值,并关闭所选的行 r_1 和列 c_1

b.转到 c_1 行,找出最接近客户 c_2 的最大值。然后将顶点 c_2 添加到路线中

c.关闭行 c_1 和列 c_2

d.重复步骤 2b,直到所有行都关闭。然后得到一个 TSP 解决方案

e.通过对每行的顶点按降序排序,从每行的质量力矩阵获得其他 n 个 TSP 解

f.考虑到容量约束(Q),所有(n+1)个 TSP 解都被转化为 CVRP 路径

g.从(n+1)条 CVRP 路线中,选择总成本最小的路线结构。

2.3 扫描算法-聚类优先路径第二启发式

该算法是一种将客户聚类成组的方法,以便同一组中的客户在地理上靠近,并且可以由同一辆车提供服务,该算法包括两个阶段,第一个阶段是聚类阶段,第二个阶段是分配路径^[19]。

假设客户是平面上的点,以欧氏距离计算。计算 (X_i, Y_i) 和 (X_j, Y_j) 之间的距离。完成该算法的步骤如下^[20]:

(1) 第一阶段:

a.计算每个客户相对于仓库的极坐标,然后通过增加极角对客户进行排序

b.向列表顶部的第一辆车添加货物,但不得超过车辆的容量。如果负载超过容量,则继续下一辆车。这一步骤一直持续到涵盖所有客户。

c.角度值 $\theta = \tan^{-1}(Y_i/X_i)$

(2) 第二阶段:

用精确或启发式方法求解每个聚类的 TSP。

这种方法可以用微软表格来解决^[21]。

3 结果

3.1 一种改进的 Clarke 和 Wright 算法—ICW 算法

在下面的表 3 中,有需要进一步处理的基本输入数据:

表 3 基本输入数据

聚类号	纬度(S)	经度(E)	X	Y	需求
DCH	-6.187692	106.773820	0.26	6.72	0
CG6	-6.126652	106.713418	7.05	0.00	0.78
CPM	-6.174611	106.790322	1.71	8.56	1.05
PRM	-6.188177	106.734230	0.20	2.32	1.71
KTA	-6.155008	106.817747	3.90	11.61	1.63
LMP	-6.190000	106.738468	0.00	2.79	0.46
GMP	-6.160906	106.818587	3.24	11.71	1.06
MTA	-6.178593	106.792792	1.27	8.84	1.12
NSF	-6.174581	106.789918	1.72	8.52	0.28

DCH: 配送中心

X 和 Y 的单位是 KM; 需求的单位是 M^3

表 4 距离矩阵

C_{ij}	城市	V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
		DCH	CG6	CPM	PRM	KTA	LMP	GMP	MTA	NSF
V_0	DCH	0	10	3	5	7	4	6	3	3
V_1	CG6	10	0	11	8	12	8	12	11	11
V_2	CPM	3	11	0	7	4	6	4	1	1
V_3	PRM	5	8	7	0	10	1	10	7	7
V_4	KTA	7	12	4	10	0	10	1	4	4
V_5	LMP	4	8	6	1	10	0	10	7	6
V_6	GMP	6	13	4	10	1	10	0	4	4
V_7	MTA	3	11	1	7	4	7	4	0	1
V_8	NSF	3	11	1	7	4	6	4	1	0

表 5 需求向量

V_i	d_i
V_1	0.78
V_2	1.05
V_3	1.71
V_4	1.63
V_5	0.46
V_6	1.06

V_7	1.12
V_8	0.28

表5中每个向量的需求是每个门店每天的平均需求。计量单位是 M^3 ，用于计算所需的卡车数量。在收集这些数据之后，现在创建节约矩阵，例如对于 S_{12} ：

$$S_{12} = C_{01} - C_{02} - C_{12}$$

$$S_{12} = 10 + 3 - 11 = 2$$

表6 节约矩阵

S_{ij}	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
V_1	0	2	7	5	6	8	2	2
V_2		0	1	5	1	5	5	5
V_3			0	2	8	1	1	1
V_4				0	1	12	6	6
V_5					0	0	0	1
V_6						0	5	5
V_7							0	5
V_8								0

表7 降序排列的节约清单

S_{ij}	节约值	S_{ij}	节约值	S_{ij}	节约值
4-6	12	2-7	5	2-5	1
1-6	8	2-8	5	3-6	1
3-5	8	6-7	5	3-7	1
1-3	7	6-8	5	3-8	1
1-5	6	7-8	5	4-5	1
2-4	6	1-2	2	5-8	1
4-7	6	1-7	2	5-6	0
4-8	6	1-8	2	5-7	0
1-4	5	3-4	2		
2-6	5	2-3	1		

现在，两阶段选择过程如下：

$T = 5$

Gene	1	2	3	4	5
S_{ij}	S_{46}	S_{16}	S_{35}	S_{13}	S_{15}
Savings (S_n)	12	8	8	7	6
P_n	0.29	0.20	0.20	0.17	0.15
Q_n	0.29	0.49	0.68	0.85	1.00

$r = 0.22$

New Gene	1
S_{ij}	S_{46}
Savings (S_n)	12

$T = 3$

Gene	1	2	3
S_{ij}	S_{16}	S_{35}	S_{13}
Savings (S_n)	8	8	7
P_n	0.35	0.35	0.30
Q_n	0.35	0.70	1.00

$r = 0.19$

New Gene	1	2
S_{ij}	S_{46}	S_{16}
Savings (S_n)	12	8

图2 使用两阶段选择程序的实例计算

重复该过程，直到表7中的所有节约被选为新基因。结果在表8中。

表8 两阶段选择过程的结果

S_{ij}	节约值	S_{ij}	节约值	S_{ij}	节约值
4-6	12	2-8	5	3-6	1
1-6	8	2-6	5	3-4	2
3-5	8	6-8	5	3-7	1
4-7	6	7-8	5	5-8	1
1-3	7	6-7	5	4-5	1
4-8	6	1-7	2	3-8	1
2-4	6	1-2	2	5-6	0
1-5	6	1-8	2	5-7	0
1-4	5	2-5	1		
2-7	5	2-3	1		

表9 路径，需求，距离-ICW

卡车编号	路径	需求(M^3)	距离(KM)
1	0-4-6-1-3-8-0; DCH - KTA - GMP - CG6 - PRM - NSF - DCH	5.46	39
2	0-2-7-5-0; DCH - CPM - MTA - LMP - DCH	2.63	15
	总计	8.09	54

3.2 Karagul Tokat Aydemir—KTA 算法

有八个客户(C_1 到 C_8)。DCH的需求为零。距离矩阵和客户需求如下表10所示。

表 10 距离矩阵和客户需求

		1	2	3	4	5	6	7	8	9
需求(q _i)			0.78	1.05	1.71	1.63	0.46	1.06	1.12	0.28
客户		DCH	C1-CG6	C2-CPM	C3-PRM	C4-KTA	C5-LMP	C6-GMP	C7-MTA	C8-NSF
1	DCH	0	10	3	5	7	4	6	3	3
2	C1-CG6		0	11	8	12	8	13	11	11
3	C2-CPM			0	7	4	6	4	1	1
4	C3-PRM				0	10	1	10	7	7
5	C4-KTA					0	10	1	4	4
6	C5-LMP						0	10	7	6
7	C6-GMP							0	4	4
8	C7-MTA								0	1
9	C8-NSF									0

然后，可以计算解决方案权重矩阵，结果如下表 11 所示。

表 11 解决方案权重矩阵

		1	2	3	4	5	6	7	8	9
需求(q _i)			0.78	1.05	1.71	1.63	0.46	1.06	1.12	0.28
客户		DCH	C1-CG6	C2-CPM	C3-PRM	C4-KTA	C5-LMP	C6-GMP	C7-MTA	C8-NSF
1	DCH	-	0.96	0.39	1.06	1.41	0.23	0.79	0.42	0.10
2	C1-CG6	0.96	0.96	0.54	0.82	0.66	0.97	0.59	0.53	0.74
3	C2-CPM	0.39	0.54	0.39	0.61	1.36	0.55	1.13	5.56	3.00
4	C3-PRM	1.06	1.82	0.61	1.06	0.60	4.79	0.54	0.60	0.67
5	C4-KTA	1.41	0.66	1.36	0.60	1.41	0.63	6.60	1.34	1.60
6	C5-LMP	0.23	0.97	0.55	4.79	0.63	0.23	0.54	0.47	0.60
7	C6-GMP	0.79	0.59	1.13	0.54	6.60	0.54	0.79	1.11	1.34
8	C7-MTA	0.42	0.53	5.56	0.60	1.34	0.47	1.11	0.42	3.00
9	C8-NSF	0.10	0.74	3.00	0.67	1.60	0.60	1.34	3.00	0.10

表 12 路径，需求，距离-KTA

实例计算：

$$X_{23} = \frac{q_2 \cdot d_2 + q_3 \cdot d_3}{(q_2 + q_3) \cdot d_{23}}$$

$$X_{23} = \frac{0.78 \cdot 10 + 1.05 \cdot 3}{(0.78 + 1.05) \cdot 11}$$

$$X_{23} = 0.54$$

当一辆汽车或卡车的最大容量为 5.5M³ 时，在实施阶段执行步骤，现在获得的路径如下表 12 所示。

卡车编号	路径	需求(M ³)	距离(KM)
1	0-9-6-8-4-7-2-0; DCH - NSF - LMP - MTA - PRM - GMP - CG6 - DCH	5.41	56
2	0-5-3-1-0; DCH - KTA - CPM - DCH	2.68	14
	总计	8.09	70

3.3 扫描算法—聚类优先路径第二

在确定集群和路径顺序之前，需要获得一些数据：theta，一辆车的最大容量，以及表 13 中的要求。

表 13 容量，坐标，需求，theta 和顺序

车辆容量	5.5	
车库编号	X	Y
0	0.26	6.72
Theta 操作		
194		

LMP	LMP	LMP	LMP	LMP	LMP	LMP
CG6	7.05	0.00	0.78	149.52	3	CG6
CPM	1.71	8.56	1.05	245.81	5	CPM
PRM	0.20	2.32	1.71	103.51	2	PRM
KTA	3.90	11.61	1.63	247.56	6	KTA
LMP	0.00	2.79	0.46	100.48	1	LMP
GMP	3.24	11.71	1.06	253.31	7	GMP
MTA	1.27	8.84	1.12	258.59	8	MTA
NSF	1.72	8.52	0.28	245.05	4	NSF

O 代表配送中心

表 14 聚类结果

顺序	1	2	3	4	5	6	7	8
聚类号	LMP	PRM	CG6	NSF	CPM	KTA	GMP	MTA
X	0.00	0.20	7.05	1.72	1.71	3.90	3.24	1.27
Y	2.79	2.32	0.00	8.52	8.56	11.61	11.71	8.84
需求	0.46	1.71	0.78	0.28	1.05	1.63	1.06	1.12
车辆号	1	1	1	1	1	2	2	2
顺序	1	2	3	4	5	1	2	3
CML-负载	0	2.17	2.95	3.23	4.28	1.63	2.69	3.81
距离	3.94	0.51	7.23	10.05	2.39	6.10	0.66	5.82
主要 VHSQ	1-1	1-2	1-3	1-4	1-5	2-1	2-2	2-3
聚类号	LMP	PRM	CG6	NSF	CPM	KTA	GMP	MTA

表 15A 优化路径顺序的结果—车辆 1

车辆号	顺序	聚类号	X	Y	距离
1	0	0	0.26	6.72	0
1	1	LMP	0.00	2.79	3.94
1	2	PRM	0.20	2.32	0.51
1	3	CG6	7.05	0.00	7.23
1	4	NSF	1.72	8.52	10.05
1	5	CPM	1.71	8.56	0.05
1	6	0	0.26	6.72	2.34

表 15B 优化路径顺序的结果—车辆 2

2	0	0	0.26	6.72	0.00
2	1	KTA	3.90	11.61	6.10
2	2	GMP	3.24	11.71	0.66
2	3	MTA	1.27	8.84	3.48
2	4	0	0.26	6.72	2.34

然后，根据表 15A 和 15B，该公司需要两辆卡车来满足需求，总距离为 36.71KM，如下表 16 所示。

表 16 路径，需求，距离 - 扫描法

卡车编号	路径	需求(M³)	距离(KM)
1	DCH - LMP - PRM - CG6 - NSF - CPM - DCH	4.28	24
2	DCH - KTA - GMP - MTA - DCH	3.81	12.58
	总计	8.09	36.58

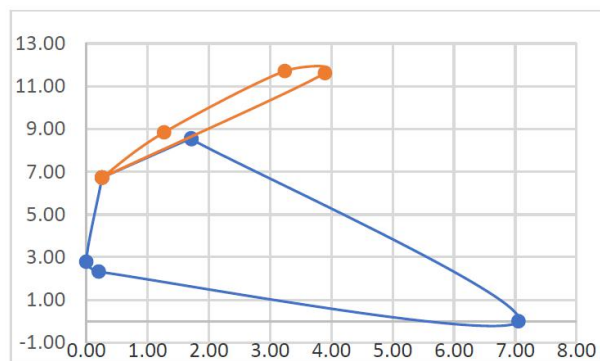


图 3 聚类图

现在，所有算法从车辆数量、路线、利用率和总距离方

面进行比较，如下表 17 所示。

表 17 算法比较

算法	车辆号	路径	需求 (M ³)	容量 (M ³)	利用率 (%)	距离 (KM)	总计 (KM)
ICW	1	DCH - KTA - GMP - CG6 - PRM - NSF - DCH	5.46	5.50	99.27%	39	54.00
	2	DCH - CPM - MTA - LMP - DCH	2.63	5.50	47.82%	15	
KTA	1	DCH - NSF - LMP - MTA - PRM - GMP - CG6 - DCH	5.41	5.50	98.36%	56	70.00
	2	DCH - KTA - CPM - DCH	2.69	5.50	48.91%	14	
扫描法	1	DCH - LMP - PRM - CG6 - NSF - CPM - DCH	4.28	5.50	77.82%	24	36.80
	2	DCH - KTA - GMP - MTA	3.81	5.50	69.27%	12.8	

		- DCH					
--	--	-------	--	--	--	--	--

所有算法都表明，要为西雅加达的所有门店执行交付，需要两辆各具有 5.5M³ 容量的车辆。ICW 算法对第一辆车的利用率最高，扫描算法对第二辆车的利用率最高。与其它两种算法相比，扫描算法的总距离最短。如果距离越短，理论上忽略拥挤因素所使用的燃料越少。扫描算法产生的总距离比 KTA 算法短 48.57%，比 ICW 算法短 33.33%。

为了证实这一发现，需要对较大样本量进行评估，例如大约 30 个样本量到 100 个样本量的中等样本量。

4 结论

与 ICW 和 KTA 算法相比，扫描算法的总距离最短，可将货物从配送中心配送到西雅加达的所有门店。所有算法都需要两辆车。需要进一步研究更大的样本量来加强这一发现。

致谢

作者感谢土耳其代尼兹利帕姆卡莱大学物流系 Kenan Karagul 教授在 KTA 算法支持材料方面的帮助。

参考文献:

- [1] K. Jerabek, P. Majercak, T. Kliestik, K. Valaskova. Application of Clark and Wright's Savings Algorithm Model to Solve Routing Problem in Supply Logistics. Preliminary communication, 2016. doi: 10.17818/NM/2016/SI7.
- [2] A. Chandra, B. Setiawan. Optimizing the Distribution Routes Using Vehicle Routing Problem (VRP) Method. Jurnal Manajemen Transportasi & Logistik, 2018,5(2):105–116.
- [3] A. Chandra, A. Naro. Studi Komparatif Metaheuristics untuk Mengoptimalkan Jalur Distribusi. Laporan Riset Internal (unpublished report). Universitas Mercu Buana; 2019.
- [4] G.D. Konstantakopoulos, S.P. Gayialis, I.P. Tatsiopoulos. Vehicle Routing Problem for Urban Freight Transportation: A Review. Conference Paper, June 2017.
- [5] T. Caric, S. Pasagic, Z. Lanovic. Vehicle Routing Problem Models. Science in Traffic Review, 2004,16(1):59–62.
- [6] F.W. Takes, W.A. Kusters. Applying Monte Carlo Techniques in the Capacitated Vehicle Routing Problem. White Paper, Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands, 2010.
- [7] T. Pichpibul, R. Kawtummachai. An Improved Clarke and Wright Savings Algorithm for the Capacitated Vehicle Routing Problem. Science Asia 2012,38:307–318.
- [8] Y. Marinakis, A. Migdalas. Annotated Bibliography in Vehicle Routing. Operations Research, An International Journal, 2007,7(1):27–46.
- [9] F.W. Takes, W.A. Kusters. Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem. Leiden Institute of

Advanced Computer Science, Leiden University, The Netherlands; 2010.

- [10] R.B.S. Shankar, K.D. Reddy, P. Venkataramaiah. Solution to a Capacitated Vehicle Routing Problem Using Heuristics and Firefly Algorithm. *International Journal of Applied Engineering Research*, 2018,13(21):15247–15254.
- [11] T. Pichpibul, R. Kawtummachai. A Heuristic Approach Based on Clarke – Wright Algorithm for Open Vehicle Routing Problem. *The Scientific World Journal*, 2013. <http://dx.doi.org/10.1155/2013/874349>.
- [12] K. Karagul, S. Tokat, F. Aydemir. Physics-inspired Optimization Algorithm for Obtaining Initial Routes of Capacitated Vehicle Routing Problem. *EURO Working Group on Vehicle Routing and Logistic Optimization (VeRoLog)*, Oslo, Norway; 2014.
- [13] K. Karagul, S. Tokat, F. Aydemir. A New Algorithm to The Construction of The Initial Routes for the Capacitated Vehicle Routing Problem. *Journal of Engineering Sciences and Design*, 2016,4(3):215–216.
- [14] N. Suthikarnnarunai. A Sweeping Algorithm for the Mix Fleet Vehicle Routing Problem. *Proceedings of International Multiconference of Engineers and Computer Scientists Volume II*, Hong Kong; 2008.
- [15] R.C. Larson, A.R. Odoni. *Urban Operations Research*. Chapter 6: Applications of Network Models. Prentice Hall, New Jersey; 1981. http://mit.edu/urban_or_book/www/book/chapter6/6.4.12.html
- [16] E. Baran. Route Determination for Capacitated Vehicle Routing Problem with Two Different Hybrid Heuristic Algorithm. *International Journal of Engineering Science and Application* 2018,2(2):55–64.
- [17] K. Karagul, S. Tokat, E. Aydemir. A New Algorithm for the Establishment of Initial Route for Capacity Constraints Vehicle Routing Problem. *Journal of Engineering Sciences and Design*, 2016,4(3):215–226. doi: 10.21923/jesd.60313.
- [18] K. Karagul, M.G. Kay, S. Tokat. A New Method for Generating Initial Solutions of Capacitated Vehicle Routing Problem. *Journal of Science* 2018,31(2):489–513.
- [19] R.F. Hartl, S.N. Parragh. *Transportation Logistics: An Introduction to Vehicle Route Problem: Cluster First Route Second Heuristics*. 2012. <https://prolog.univie.ac.at/teaching/LVAs/KFKTL/WS%2012/TL-Part5-VRP-Intro-Handout.pdf>
- [20] R.B.S. Shankar, K.D. Reddy. A Comparative Study on Heuristic and Metaheuristic Approach in Solving a Capacitated Vehicle Routing Problem. *International Journal of Innovative Science and Research Technology* 2018,3(9):94–98.
- [21] B. Wibowo. Budhi's Notes. Part II – Solving Vehicle Routing Problem with Excel. 2016. <https://budhiwibowo.files.wordpress.com/2016/03/dss-case-2-vehicle-routing-problem.xlsx>.