APPLICATION OF A GENETIC ALGORITHM FOR SOLVING TRAVELING SALESMAN PROBLEM IN ORGANIC PORRIDGE DISTRIBUTION

H. R. Finan, M.T. Julianto, and *E. Khatizah

School of Data Science, Mathematics, and Informatics, Bogor Agricultural University
Jl. Meranti, Kampus IPB Dramaga, Bogor.
hauralia_finr@apps.ipb.ac.id, mtjulianto@apps.ipb.ac.id,
elis_khatizah@apps.ipb.ac.id ***corresponding author**

Abstract

This study focuses on determining an optimal distribution route for organic porridge products produced by a company and delivered to multiple outlets. Each outlet is visited exactly once, and the delivery process starts and ends at the same outlet. A total of 44 outlets are considered, which are initially divided into nine distribution routes. To improve distribution efficiency, this study proposes reorganizing the outlets into only three distribution routes. Each route formulation is modeled as a Traveling Salesman Problem (TSP). The optimization of the three TSP cases is carried out using a Genetic Algorithm (GA). In the GA implementation, the order of outlets along a route is encoded as a chromosome consisting of a sequence of genes. The fitness function is defined based on the total travel distance, where a smaller value indicates a better solution. The results show that increasing the number of iterations and the size of population, which is the number of candidate routes considered at each step, can reduce the total travel distance up to a certain point. The exact routes and their sequence of outlets can be visualized in a map depicting each of the three optimized paths.

Keywords: Genetic Algorithm, distribution routing, total distance, Traveling Salesman Problem

1 Introduction

Distribution is the process of delivering products or goods from one location to another and plays a crucial role in the overall production system. Effective distribution is especially important for food products, as it directly affects product quality and freshness until the goods reach consumers. Therefore, food producers must carefully plan their distribution activities to ensure that products are delivered efficiently and in good condition.

One important aspect of an efficient distribution system is the determination of delivery routes with optimal travel distances. By selecting appropriate routes, producers can reduce transportation time and cost, while also maintaining product quality. However, many food product manufacturers still face difficulties in determining distribution routes that minimize total travel distance, particularly when the number of delivery locations continues to increase.

2020 Mathematics Subject Classification: 90C27, 68T20.

Diajukan: 25/12/2025, diterima: 29/12/2025. DOI: https://doi.org/10.29244/milang.21.2.101-116

The problem of determining an optimal route that visits each location exactly once and returns to the starting point can be modeled as a Traveling Salesman Problem (TSP). The TSP is a well-known NP-complete problem, meaning that there is no exact algorithm specifically designed to solve it efficiently for large-scale cases [12]. As a result, heuristic and metaheuristic approaches are commonly applied to obtain near-optimal solutions within a reasonable computation time.

Genetic Algorithm (GA) is one of the metaheuristic methods that can be used to solve complex optimization problems. Inspired by Darwin's theory of evolution, GA is a population-based algorithm that explores a wide and complex search space through processes such as selection, crossover, and mutation. Due to its flexibility and strong global search capability, GA has been successfully applied to various optimization problems, including routing, scheduling, and manufacturing optimization [5, 9].

GA has also been applied in areas beyond routing, such as job-shop scheduling, feature selection in machine learning, and engineering design optimization, showing its robustness in exploring large, complex search spaces [1, 13]. Meanwhile, TSP itself has been extensively studied in logistics, transportation planning, and supply chain management, often solved with metaheuristic approaches like Ant Colony Optimization, Simulated Annealing, Particle Swarm Optimization, or hybrid GA methods [2, 4, 7]. These studies demonstrate the effectiveness of GA and other metaheuristics in providing near-optimal solutions, especially in large-scale or real-world problems where exact methods are impractical.

Despite these advances, most previous research either relies on benchmark datasets or focuses on theoretical improvements, with limited application to real distribution systems of small and medium-sized enterprises (SMEs) in the food industry. In practice, companies often use predetermined or manually optimized routes, which may become inefficient as the number of outlets increases. This gap highlights the need for applied studies that demonstrate how GA-based TSP solutions can improve operational efficiency in real-world distribution settings.

In this study, the Traveling Salesman Problem (TSP) in the distribution of organic porridge products is addressed using a Genetic Algorithm (GA). The case study focuses on an organic porridge product produced by a manufacturing company in Bojonggede, Indonesia. Currently, the outlets are spread across several major cities, including Bogor City, Bogor Regency, and Depok City. As a rapidly growing SME, the company adds new outlets almost every month, which increases the complexity of planning efficient distribution routes. We focus on reorganizing the distribution routes to improve operational efficiency and on determining an optimal delivery route with the minimum total travel distance by applying a GA-based solution for the TSP. This approach aims to provide a practical optimization method suitable for real-world SME distribution challenges. The expected benefits of the research are to support decision-making for distribution planning and to enhance the overall operational performance of the company.

2 Data and Method

The organic porridge manufacturing company provided data during the author's internship, covering 44 outlets organized into 9 distribution routes for the period March to April 2021. The dataset includes the number of outlets, the number of distribution routes, the location of each outlet, and the distances between outlets. A complete list of the distribution routes and their associated outlets is presented in Table 1.

Table 1. Company outlets and its distribution routes for March–April 2021.

No.	Outlet name	Distribution route	Outlet code
1	Taman Citayam Elok	Citayam	TCE
2	Ragajaya	Citayam	RJ
3	Grand Hills	Citayam	GH
4	Bukit Pesanggrahan Indah	Citayam	BPI
5	Tugu Macan	Citayam	TM
6	Puri Nirwana 3	Cibinong	PN3
7	Sukahati 2	Cibinong	SH2
8	Gaperi 2	Cibinong	GP2
9	Kaum Pandak	Cibinong	KP
10	Griya Cibinong Indah	Cibinong	GCI
11	Pakansari	Cilodong	PKS
12	Al-Baaliah	Cilodong	AB
13	Al-Falah	Cilodong	AF
14	Pedurenan	Cilodong	PDN
15	Cilodong	Cilodong	CLD
16	Jambu Dipa	Mandala	JD
17	Villa Bogor Indah 2	Mandala	VBI
18	Patung Kuda	Mandala	PKU
19	Mandala	Mandala	MDL
20	Pasir Jambu	Mandala	PSJ
21	Kedung Badak	Kebon Pedes	KB
22	Pondok Rumput	Kebon Pedes	PDR
23	Teplan	Kebon Pedes	TPL
24	Kebon Pedes	Kebon Pedes	BDS
25	Gang Makam	Cilendek	GM
26	Cilendek Timur	Cilendek	CT
27	Cimanggu Poncol	Cilendek	CP
28	Cilendek	Cilendek	CLK
29	Gang Swadaya	Depok	GSW
30	Lipi Pondok Rajeg	Depok	LPR
31	Kalimulya	Depok	KM
32	Jati Mulya	Depok	JTM
33	Palapa	Depok	PLP
34	Sukmajaya Depok	Depok	SJD
35	Pasar Ciluar	Ciluar	PCR
36	Pasirlaja	Ciluar	PSL
37	Tanah Baru	Ciluar	TB
38	Sogiri	Ciluar	SGR
39	Tanah Baru Indah	Ciluar	TBI
40	Ciampea	Ciampea	CMP
41	Puri Arraya 1	Ciampea	AR1
42	Puri Arraya 2	Ciampea	AR2
43	Taman Dramaga Permai	Ciampea	TDP
44	Griya Dramaga Asri	Ciampea	GDA

To apply a Genetic Algorithm (GA) for solving the distribution problem from the company to its outlets, the outlets are initially regrouped into new distribution routes. After this step, three new distribution routes are created, and the route for each distribution track is determined using the Traveling Salesman Problem (TSP) framework. GA is then applied to each of the three new distribution routes to find the optimal visiting sequence of outlets.

During the GA implementation, parameters such as the number of iterations and population size are adjusted to obtain the best possible results. Finally, the outcomes of GA for solving the TSP are recorded and analyzed to evaluate the efficiency of the proposed distribution routes. This approach ensures that the distribution planning is practical and effective for the real-world conditions of a growing SME.

2.1 Reorganizing Distribution Routes

In this study, the actual distribution system at the company consists of nine distribution routes, which are Citayam, Cibinong, Cilodong, Mandala, Kebon Pedes, Cilendek, Depok, Ciluar, and Ciampea. The distribution is currently carried out using motorcycles. To improve efficiency, the nine existing routes are regrouped into three new distribution routes, referred to as Distribution Lines. This reorganization not only aims to optimize route planning but also simulates a scenario in which a large number of new outlets is added to the system. It is also suggested that motorcycles be replaced with vehicles capable of carrying more goods safely.

The three new distribution routes were formed by consolidating the nine original routes into three geographically coherent lines. No formal algorithm was used, since the regrouping was based on practical considerations. Solving each of the original nine routes individually using the GA-TSP would have been too simple, as each route contains only 4–6 outlets, as shown in the Distribution Route column of Table 1. To make the simulation meaningful, the nine routes were reorganized into three new routes, each consisting of three former routes. The selection of routes to combine was guided by geographical proximity within the Bogor area, as visualized using Google Maps. Specifically, Distribution Line A combines Kebon Pedes, Cilendek, and Ciampea in the southwest; Distribution Line B combines Mandala, Cibinong, and Citayam in the central area; and Distribution Line C combines Ciluar, Cilodong, and Depok vertically elongated in the east. The full list of outlets for each distribution line is presented in Table 2 (column Distribution Line).

The distances between each pair of outlets are measured using Google Maps and organized into distance matrices for each route, shown separately in Tables 3, 4, and 5. These distance matrices are used as input for the Genetic Algorithm to determine the optimal visiting sequence that minimizes the total travel distance on each route.

The model used in this study is based on several simplifying assumptions. First, the distances between outlets are assumed to be symmetric, meaning the distance from outlet A to outlet B is the same as from outlet B to outlet A. Second, each distribution line is served by a single vehicle, so the problem is treated as a standard TSP for each line. Third, the distribution is assumed to occur within a single period, requiring all outlets to be visited in one continuous trip. These assumptions allow the study to focus on the application of the Genetic Algorithm to determine the optimal sequence of outlets along each route.

Table 2. List of outlet names on distribution line A, B and C.

No.	Outlet name	Distribution route	Distribution line
1	Taman Citayam Elok	Citayam	B11
2	Ragajaya	Citayam	B12
3	Grand Hills	Citayam	B13
4	Bukit Pesanggrahan Indah	Citayam	B14
5	Tugu Macan	Citayam	B15
6	Puri Nirwana 3	Cibinong	B6
7	Sukahati 2	Cibinong	B7
8	Gaperi 2	Cibinong	B8
9	Kaum Pandak	Cibinong	B9
10	Griya Cibinong Indah	Cibinong	B10
11	Pakansari	Cilodong	C6
12	Al-Baaliah	Cilodong	C7
13	Al-Falah	Cilodong	C8
14	Pedurenan	Cilodong	C9
15	Cilodong	Cilodong	C10
16	Jambu Dipa	Mandala	B1
17	Villa Bogor Indah 2	Mandala	B2
18	Patung Kuda	Mandala	B3
19	Mandala	Mandala	B4
20	Pasir Jambu	Mandala	B5
21	Kedung Badak	Kebon Pedes	A1
22	Pondok Rumput	Kebon Pedes	A2
23	Teplan	Kebon Pedes	A3
24	Kebon Pedes	Kebon Pedes	A4
25	Gang Makam	Cilendek	A5
26	Cilendek Timur	Cilendek	A6
27	Cimanggu Poncol	Cilendek	A7
28	Cilendek	Cilendek	A8
29	Gang Swadaya	Depok	C11
30	Lipi Pondok Rajeg	Depok	C12
31	Kalimulya	Depok	C13
32	Jati Mulya	Depok	C14
33	Palapa	Depok	C15
34	Sukmajaya Depok	Depok	C16
35	Pasar Ciluar	Ciluar	C1
36	Pasirlaja	Ciluar	C2
37	Tanah Baru	Ciluar	C3
38	Sogiri	Ciluar	C4
39	Tanah Baru Indah	Ciluar	C5
40	Ciampea	Ciampea	A9
41	Puri Arraya 1	Ciampea	A10
42	Puri Arraya 2	Ciampea	A11
43	Taman Dramaga Permai	Ciampea	A12
44	Griya Dramaga Asri	Ciampea	A13

Because each distribution line is served by a single distribution vehicle, the distribution process on each line can be modeled as a graph $G = (V, E)$ with $v \in V$ represents an outlet and each edge $e = (i, j) \in E$ represents the path connecting outlet v_i

and outlet v_j. The cost of visiting $c(i,j)$ corresponds to the distance between the two outlets. Let V_A, V_B, and V_C denote the sets of outlets in Distribution lines A, B, and C, and E_A, E_B, and E_C denote the sets of edges connecting outlet pairs in each line. The distribution of organic porridge on Routes A, B, and C can thus be modeled as a Traveling Salesman Problem on graphs $G_A = (V_A, E_A)$, $G_B = (V_B, E_B)$ and $G_C = (V_C, E_C)$ respectively.

Table 3. Distance matrix between outlets on Distribution Line A, in kilometers

Outlet	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
A1	0	1.7	0.7	1	0.95	3.5	2	3.6	15	16	20	12	13
A2	1.7	0	1.8	0.65	1.3	3.9	2.3	4.5	16	17	20.5	12	13
A3	0.7	1.8	0	1.1	1.9	5.9	3	5.8	16.5	17.5	21	13	13.1
A4	1	0.65	1.1	0	1.4	4.3	3.3	5	16	17	21	14	14
A5	0.95	1.3	1.9	1.4	0	3.3	1.7	3.2	14	15	19	12	12
A6	3.5	3.9	5.9	4.3	3.3	0	1.8	1.5	12	13	17	8.9	9.8
A7	2	2.3	3	3.3	1.7	1.8	0	1.7	14	15	19	12	12
A8	3.6	4.5	5.8	5	3.2	1.5	1.7	0	12	13	17	9.6	9.7
A9	15	16	16.5	16	14	12	14	12	0	3.5	7.3	5.4	2.5
A10	16	17	17.5	17	15	13	15	13	3.5	0	7	6.5	3.6
A11	20	20.5	21	21	19	17	19	17	7.3	7	0	10	7.5
A12	12	12	13	14	12	8.9	12	9.6	5.4	6.5	10	0	3.4
A13	13	13	13.1	14	12	9.8	12	9.7	2.5	3.6	7.5	3.4	0

Table 4. Distance matrix between outlets on Distribution Line B, in kilometers

Outlet	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
B1	0	3.4	1.6	4.3	2.5	4.1	6.5	6.2	2.9	6.6	9.1	11	13	10	10
B2	3.4	0	1.8	1.9	1.3	5.4	7.7	7.5	4.1	6.5	13	15	16	14	15
B3	1.6	1.8	0	2.9	0.95	3.2	5.5	5.3	1.9	5.6	11	12	14	12	13
B4	4.3	1.9	2.9	0	2	5.1	7.3	7.2	4	5.4	13	15	17	15	15
B5	2.5	1.3	0.95	2	0	4	6.4	6.1	2.8	6.5	11	13	15	13	14
B6	4.1	5.4	3.2	5.1	4	0	2.7	2.4	0.55	3.8	11	12	13	10	10
B7	6.5	7.7	5.5	7.3	6.4	2.7	0	1.6	3.2	6.5	7.9	9	10.9	8.3	7.5
B8	6.2	7.5	5.3	7.2	6.1	2.4	1.6	0	2.2	5.5	9.4	11	12	9.9	9
B9	2.9	4.1	1.9	4	2.8	0.55	3.2	2.2	0	3.3	12	13	15	13	13
B10	6.6	6.5	5.6	5.4	6.5	3.8	6.5	5.5	3.3	0	12	14	15	12	12
B11	9.1	13	11	13	11	11	7.9	9.4	12	12	0	1.7	2	0.8	1.6
B12	11	15	12	15	13	12	9	11	13	14	1.7	0	0.4	1.2	1.7
B13	13	16	14	17	15	13	10.9	12	15	15	2	0.4	0	1.6	2.1
B14	10	14	12	15	13	10	8.3	9.9	13	12	0.8	1.2	1.6	0	0.45
B15	10	15	13	15	14	10	7.5	9	13	12	1.6	1.7	2.1	0.45	0

Table 5. Distance matrix between outlets on Distribution Line C, in kilometers

Outlet	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
C1	0	0.25	1.2	1.3	4.3	7.9	11	11	10.9	14	13	13.1	16	15	15	18
C2	0.25	0	1.2	3	4.4	8.1	12	13	12	16	14	15	18	16	17	19
C3	1.2	1.2	0	1.8	3.1	8	11	11	11	14	13	14	18	15	17	18
C4	1.3	3	1.8	0	1.6	13	17	16	16	19	17	18	21	19	22	19
C5	4.3	4.4	3.1	1.6	0	14	18	17	17	22	19	19.1	22	20	23	21
C6	7.9	8.1	8	13	14	0	5.1	5.9	5	7.8	4.2	5	9.3	6.1	8.3	11
C7	11	12	11	17	18	5.1	0	2.9	1.1	5.9	3.6	4.4	7.3	4.4	6.2	8.8
C8	11	13	11	16	17	5.9	2.9	0	4.7	2.1	2.9	2.8	5.2	2.7	4.1	6.7
C9	10.9	12	11	16	17	5	1.1	4.7	0	4.9	4.6	4.6	6.9	4.4	5.8	8.4
C10	14	16	14	19	22	7.8	5.9	2.1	4.9	0	4.8	4	3	1.2	3.3	3.2
C11	13	14	13	17	19	4.2	3.6	2.9	4.6	4.8	0	0.2	5.2	1.9	4.1	7.1
C12	13.1	15	14	18	19.1	5	4.4	2.8	4.6	4	0.2	0	5.4	2.1	4.3	7.3
C13	16	18	18	21	22	9.3	7.3	5.2	6.9	3	5.2	5.4	0	3.7	1.1	1.7
C14	15	16	15	19	20	6.1	4.4	2.7	4.4	1.2	1.9	2.1	3.7	0	2.6	5.2
C15	15	17	17	22	23	8.3	6.2	4.1	5.8	3.3	4.1	4.3	1.1	2.6	0	2.8
C16	18	19	18	19	21	11	8.8	6.7	8.4	3.2	7.1	7.3	1.7	5.2	2.8	0

2.2 Implementation of the Genetic Algorithm (GA) for Solving TSP

The process of applying the GA to solve the TSP begins with initializing the GA parameters: the distance matrix between outlets (in kilometers), the number of iterations I (dimensionless, representing the number of generations), the crossover rate P_c (dimensionless, representing the probability of crossover between two parent chromosomes), the mutation rate P_m (dimensionless, representing the probability of mutation for a gene), and population size N (dimensionless, representing the number of candidate chromosomes per generation). After the population is generated, the GA proceeds through several steps: evaluating and selecting chromosomes based on fitness values, performing crossover and mutation operations, and regenerating the population for the next iteration. A brief technical explanation of the GA procedure is provided in the Appendix.

The study uses discrete decimal encoding, where each gene represents an outlet. If there are n outlets, the genes are integers from 1 to n. A route of outlets forms a chromosome, and a group of chromosomes is called a population or a set of candidate solutions [10]. The population size, denoted as N, can be set as needed. For example, with 13 outlets, each chromosome consists of 13 genes, and the population contains N chromosomes.

Suppose the k-th chromosome is denoted as z_k with $k = 1,2,\cdots,N$ consisting of a sequence of genes denoted v_t with $t = 1,2,\cdots,n$, then we have $z_k = v_1, v_2, \cdots, v_n$. An illustration of a population with chromosomes of $n = 13$ genes is shown in Figure 1.

Gene

z_1	1	8	7	5	2	3	4	6	10	9	12	11	13
z_2	5	11	1	7	3	2	4	6	9	13	10	8	12
$\vdots$													
z_k	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}
$\vdots$													
z_N	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}

Figure 1. The illustration of a population with $n = 13$ chromosomes.

Once the chromosomes are generated, GA parameters are set, consisting of I, P_c, P_m, and N. Larger populations and more iterations allow the algorithm to explore more solutions and improve the chances of finding the best route.

The algorithm then evaluates the fitness of each chromosome, calculated as the total travel distance of the route, with smaller distances indicating better solutions. The fitness of a chromosome z_k is calculated using Equation (1)

$$f(z_k) = \left(\sum_{t=1}^{n-1} c(v_t, v_{t+1})\right) + c(v_n, v_1), \tag{1}$$

where $c(i, j)$ is the distance between *outlet* i and *outlet* j. The goal is to achieve the smallest fitness value, which represents the TSP route with the minimal total travel distance. For example, the fitness values for chromosomes z_1 and z_2 based on Figure 1 are as follow.

$$\begin{aligned}
f(z_1) &= \left(\textstyle\sum_{t=1}^{12} c(v_t, v_{t+1})\right) + c(v_{13}, v_1) \\
&= 3.6 + 1.7 + 1.7 + 1.3 + 1.8 + 1.1 + 4.3 + 13 + 3.5 + 5.4 + 10 + 7.5 + 13 \\
&= 54.9, \\
f(z_2) &= \left(\textstyle\sum_{t=1}^{12} c(v_t, v_{t+1})\right) + c(v_{13}, v_1) \\
&= 3.2 + 19 + 20 + 3 + 1.8 + 0.65 + 4.3 + 12 + 2.5 + 3.6 + 7 + 13 + 5.4 \\
&= 95.45.
\end{aligned}$$

Chromosomes with the best fitness are selected as parents for one-point crossover, producing new offspring while avoiding duplicate genes.

To determining the chromosome that has the best fitness value after calculating their fitness values, selection is done proportionally based on fitness using the roulette wheel method. In this method, each chromosome occupies a section of the roulette wheel proportional to its fitness value [6]. Fitness values are updated continuously until the set population size N and the maximum number of iterations are reached. Chromosomes with the smallest fitness values have a higher chance of being selected as parents. For example, in distribution line A, chromosome z_1 with the sequence $1-8-7-5-2-3-4-6-10-9-12-11-13$ has a fitness of 54.9, while chromosome z_2 with the sequence $5-11-1-7-3-2-4-6-9-13-10-8-12$ has a fitness of 95.45.

The two best chromosomes selected as parents are then crossed over using the crossover process. A random number $r \in (0,1)$ is generated, and if r is less than or equal to the initialized crossover rate $(r \leq P_c)$, crossover is performed. For example, if P_c is set to 1, crossover occurs for every pair of selected parents. The one-point crossover method is used, producing two new offspring chromosomes from the parent chromosomes. In this method, a cut point (CP) is first determined, for example, for Distribution Line A, the CP is calculated as shown in Equation (2).

$$CP = \left\lceil \frac{n}{2} \right\rceil = \left\lceil \frac{13}{2} \right\rceil = 7. \tag{2}$$

The substrings from the parents to be crossed are selected based on this cut point. The selected segments of the parent chromosomes are then exchanged to create the offspring, ensuring that no gene appears more than once in a chromosome. Any missing genes are inserted to complete the chromosomes. An illustration of the one-point crossover process is shown in Figure 2. The resulting offspring chromosomes from the one-point crossover process are then ready to undergo mutation.

Parent chromosome z_1	1	8	7	5	2	3	4	6	10	9	12	11	13
Parent chromosome z_2	5	11	1	7	3	2	4	6	9	13	10	8	12

Offspring chromosome z_1	1	8	7	5	2	3	4	6	9	13	10		12
Offspring chromosome z_2	5	11	1	7	3	2	4	6	10	9	12		13

Offspring chromosome z_1	1	8	7	5	2	3	4	6	9	13	10	11	12
Offspring chromosome z_2	5	11	1	7	3	2	4	6	10	9	12	8	13

Figure 2. The example of one-point crossover process in Distribution Line A.

Generally, after crossover process, offspring may undergo mutation, which randomly swaps two genes to explore new solutions and prevent convergence to local optima. A random number $r \in (0,1)$ is generated, and if $r \leq P_m$, the chromosome undergoes mutation. For example, if P_m is set to 0.9, 90% of the genes in the population are subject to mutation. The swap mutation technique exchanges two randomly selected genes in the offspring chromosome, as illustrated in Figure 3. Regeneration then replaces the worst chromosomes with the new offspring, maintaining a constant population size.

Offspring chromosome z_1 before mutation	1	8	7	5	2	3	4	6	9	13	10	11	12
Offspring chromosome z_1 after mutation	1	8	3	5	2	7	4	6	9	13	10	11	12

Offspring chromosome z_2 before mutation	5	8	1	7	3	2	4	6	10	9	12	11	13
Offspring chromosome z_2 after mutation	5	4	1	7	3	2	8	6	10	9	12	11	13

Figure 3. The example of swap mutation technique in Distribution Line A.

This process, covering evaluation, selection, crossover, mutation, and regeneration, repeats until the maximum number of iterations is reached. Different population sizes and iteration numbers are tested to compare results and computation times. The final output is the chromosome representing the shortest route for each distribution path.

3 Result and Discussion

The Genetic Algorithm for the TSP was implemented using Octave 6.2.0 on a computer with an Intel Celeron N4000 processor, 4 GB RAM, and Microsoft Windows 10 Professional 64-bit. The programming code was adapted from Atiyatna [14] with modifications to suit this study. The parameters used were based on previous studies in [8, 11], the crossover rate (P_c) was set to 1, the mutation rate (P_m) to 0.9, the number of iterations (I) to $100, 150, 200$, and 250, and the population size (N) to $50, 100, 150$, and 200.

After performing the GA procedures described in the method section, the results included the minimum total distance and computation time for each set of parameters. Since GA is a stochastic method, the results may vary between runs. Therefore, the program was executed multiple times for each parameter set, and the best total distance and average computation time were recorded.

3.1 GA Implementation on Distribution Line A, B, and C

The GA implementation for Distribution Line A, B and C, is summarized in Table 6, Table 7, and Table 8. A total of 12 experiments were conducted for each distribution line with the parameters set to $P_c = 1$ and $P_m = 0.9$. In the first four experiments, the number of iterations (I) was increased while the population size (N) remained constant. In the next four experiments, only the population size (N) was increased. The last four experiments combined specific increases in both the number of iterations (I) and the population size (N).

Table 6. GA implementation on Distribution Line A with different parameters

No	Parameter				Minimum total distance (kilometers)	Computation time (seconds)
	I	N	P_c	P_m		
1	100	50	1	0.9	52.80	0.12
2	150	50	1	0.9	52.45	0.18
3	200	50	1	0.9	52.30	0.23
4	250	50	1	0.9	52.30	0.24
5	100	50	1	0.9	52.80	0.12
6	100	100	1	0.9	52.60	0.14
7	100	150	1	0.9	53.20	0.16
8	100	200	1	0.9	52.30	0.17
9	100	50	1	0.9	52.80	0.12
10	150	100	1	0.9	52.85	0.20
11	200	150	1	0.9	52.30	0.26
12	250	200	1	0.9	52.25	0.34

As shown in Tabel 6, the minimum total distance for Distribution Line A was 52.25 km, with a computation time of 0.34 seconds, using 250 iterations ($I = 250$) and a population size of 200 ($N = 200$). The route with this minimal distance is A3 – A4 – A2 – A5 – A7 – A8 – A13 – A9 – A10 – A11 – A12 – A6 – A1. Increasing only the number of iterations from 100 to 250 reduced the distance from 52.80 km to 52.30 km, demonstrating that higher iteration counts allow the GA to explore more solutions and converge closer to the optimal route. Changes in population size had a similar effect on distance, although computation time increased slightly less than it did with more iterations. Furthermore, the results suggest that the optimal combination of iteration number and population size can achieve the shortest distance, even if it requires the highest computation time.

Referring to Table 7, a similar pattern was observed for Distribution Line B. However, increasing only the number of iterations was sufficient to achieve the minimum total distance of 39.65 km, using 250 iterations and a population size of 50. The corresponding route is B2 – B5 – B3 – B1 – B11 – B13 – B12 – B14 – B15 – B7 – B8 – B6 – B9 – B10 – B4. The optimal combination of iteration number and population size can also achieve the shortest distance, but it requires more computation time. On the other hand, Table 8 displays that Distribution Line C achieved the minimum total distance of 47.40 km (0.48 seconds) with the same number of iterations and population size as Line B, along the route C2 – C6 – C11 – C12 – C14 – C15 – C13 – C16 – C10 – C8 – C7 – C9 – C1 – C4 – C5 – C3.

Taken together, these results indicate that GA demonstrates efficient performance for solving the TSP in small-to medium-sized distribution networks. Increasing the number of iterations has a stronger impact on solution quality than increasing the population size, while computation times remain very low across all tested scenarios. These findings indicate that GA is suitable for practical route optimization tasks, providing fast and near-optimal solutions even when multiple parameters are varied.

Table 7. GA implementation on Distribution Line B with different parameters

No	Parameter				Minimum total distance (kilometers)	Computation time (seconds)
	I	N	P_c	P_m		
1	100	50	1	0.9	44.10	0.33
2	150	50	1	0.9	45.35	0.47
3	200	50	1	0.9	42.50	0.60
4	250	50	1	0.9	39.65	0.75
5	100	50	1	0.9	44.10	0.30
6	100	100	1	0.9	43.80	0.31
7	100	150	1	0.9	42.15	0.33
8	100	200	1	0.9	42.10	0.45
9	100	50	1	0.9	44.10	0.33
10	150	100	1	0.9	44.65	0.39
11	200	150	1	0.9	41.85	0.43
12	250	200	1	0.9	39.65	1.03

Table 8. GA implementation on Distribution Line C with different parameters

No	Parameter				Minimum total distance (kilometers)	Computation time (seconds)
	I	N	P_c	P_m		
1	100	50	1	0.9	52.35	0.20
2	150	50	1	0.9	54.40	0.27
3	200	50	1	0.9	54.20	0.36
4	250	50	1	0.9	47.40	0.48
5	100	50	1	0.9	52.35	0.20
6	100	100	1	0.9	54.85	0.21
7	100	150	1	0.9	49.00	0.22
8	100	200	1	0.9	48.65	0.29
9	100	50	1	0.9	52.35	0.20
10	150	100	1	0.9	54.15	0.34
11	200	150	1	0.9	48.65	0.35
12	250	200	1	0.9	48.00	0.40

3.2 Distribution Line Map

Based on the results for Distribution Lines A, B, and C, the optimal distribution routes with minimal total distances are illustrated in Figure 4. In the figure, the red route represents Distribution Line A, the blue route represents Distribution Line B, and the green route represents Distribution Line C.

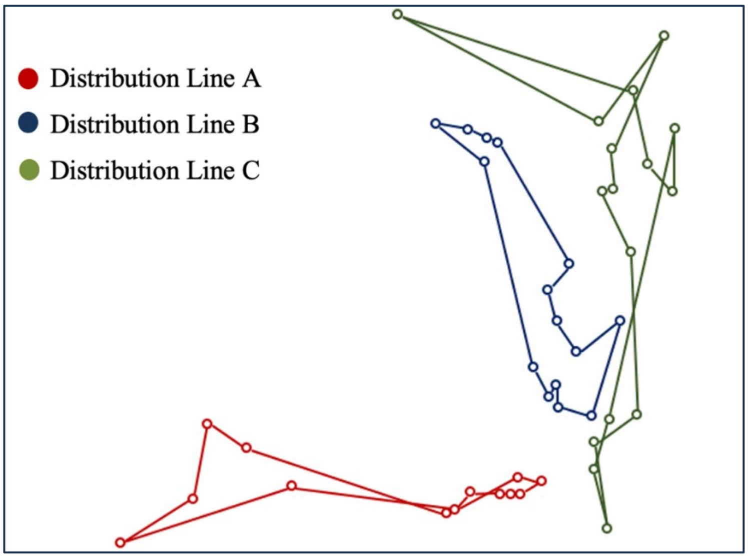

Figure 4. Distribution Lines map based on the minimal total distances.

Figure 4 shows clear visual differences among the three distribution lines that help explain the numerical results. Distribution Line A (red) forms a compact route, indicating that most of its outlets are closely located. Distribution Line B (blue) exhibits a relatively simple route shape, which is consistent with its lower total distance. In contrast, Distribution Line C (green) covers a wider area and displays sharper directional changes along its route. This characteristic helps explain why increasing the number of iterations does not always lead to consistent improvements in total distance for Distribution Line C, as observed in the numerical results. Overall, the visual patterns in Figure 9 support the quantitative findings reported in Tables 6-8.

It is also important to note that solutions obtained using genetic algorithms are not guaranteed to appear visually smooth or intuitive [3]. A route that looks elongated, irregular, or even crossing on a map may still be optimal under the traveling salesman problem objective, which focuses solely on minimizing total distance. Therefore, visual complexity does not necessarily indicate suboptimality. From Figure 4, the green route corresponding to Distribution Line C appears more spatially spread and vertically elongated, while Distribution Line A (red) appears more compact and clustered. This visual contrast may initially raise doubts, as one might expect Distribution Line A to cover a larger area. However, this difference is consistent with the underlying data characteristics rather than a labeling error. Each distribution line is optimized independently using the genetic algorithm based on its own set of outlets, and the outlets in Distribution Line C are naturally more geographically dispersed than those in Distribution Line A. Since the genetic algorithm minimizes total distance rather than geographical compactness, a route may appear long or irregular on the map but still be optimal in terms of distance.

4 Conclusion

During the March–April 2021 period, organic porridge products from a company in Bogor was distributed to 44 outlets across nine routes, which were not systematically organized and were determined by couriers during delivery. This study proposes consolidating the outlets into three distribution routes, referred to as Distribution Lines A, B, and C, with route sequences optimized using a Genetic Algorithm (GA) for the Traveling Salesman Problem (TSP). Distances between outlets were measured using Google Maps.

The proposed approach successfully generated optimal routes based on a series of experiments. The choice of GA parameters, which are number of iterations (I) and population size (N), significantly affects the results, with iterations having a slightly greater impact. The minimal total travel distances achieved for Distribution Lines A, B, and C were 52.25 km ($I = 250, N = 200$), 39.65 km ($I = 250, N = 50$), and 47.4 km ($I = 250, N = 50$), respectively, and computation times were all under 1 second.

A limitation of this study is that the original or factual distribution routes used by the company are not available, so the GA-optimized routes cannot be directly compared with actual routes. In addition, the study did not consider distribution constraints such as vehicle capacity, route restrictions, or delivery costs. Future research should incorporate these constraints to further refine route optimization and ensure practical applicability in real-world distribution scenarios.

Appendix

According to [6], the steps of the Genetic Algorithm (GA) can be summarized as follows:

1. Encoding and parameter setup
   Before applying GA, the problem must be converted into individuals represented as one or more chromosomes. Gene encoding can use several schemes:
   - Real number encoding: each gene is a real value within the interval [0, R], usually R = 1.
   - Discrete decimal encoding: each gene is represented as an integer within a defined range.
   - Binary encoding: each gene is represented as 0 or 1.

   In addition, GA parameters are defined:
   Number of iterations (I): the total number of generations the algorithm runs.
   Population size (N): the number of chromosomes in each generation.
   Crossover rate (P_c): the probability that two parent chromosomes exchange genes during crossover.
   Mutation rate (P_m): the probability that a gene in a chromosome mutates.
2. Population initialization
   The initial population of (N) chromosomes is generated, each representing a candidate solution.
3. Evaluation and selection
   Each chromosome is evaluated using a fitness function, which measures the quality of the solution (e.g., total travel distance). Chromosomes with higher fitness have a higher chance of being selected as parents and surviving to the next generation.
4. Crossover
   Crossover is performed on selected parent chromosomes with probability (P_c) producing offspring by exchanging gene segments. A higher crossover rate increases exploration of the solution space and helps reach the optimal solution faster.

5. Mutation
   Mutation is applied with probability (P_m) randomly changing genes in offspring chromosomes. This introduces diversity and prevents premature convergence to local optima.
6. Regeneration
   Offspring from crossover and mutation replace the worst chromosomes in the population, keeping the population size constant.
7. Iteration
   Steps 3–6 are repeated for *I* generations until the maximum number of iterations is reached.
8. Output
   After completing *I* generations, the chromosome with the highest fitness is selected as the optimal solution.

Referrences

[1] Alexander and H. Sriwindono, “The omparison of Genetic Algorithm and Ant Colony Optimization in completing traveling salesman problem,” in *Proceedings of the 2nd International Conference of Science and Technology for the Internet of Things, ICSTI 2019, September 3rd 2019, Yogyakarta, Indonesia*, EAI, 2020. doi: 10.4108/eai.20-9-2019.2292121.

[2] A. H. M. S. Islam, M. Tanzim, S. Afreen, and G. Rozario, “Evaluation of Ant Colony Optimization Algorithm compared to Genetic Algorithm, dynamic programming and Branch and Bound Algorithm regarding traveling salesman problem,” *Global Journal of Computer Science and Technology*, vol. 19, no. D3, pp. 7–12, 2019.

[3] A. Vie, A. M. Kleinnijenhuis, and D. J. Farmer, “Qualities, challenges and future of genetic algorithms: a literature review,” arXiv:2011.05277 [cs, math], Sep. 2021, Available: https://arxiv.org/abs/2011.05277.

[4] G. Chen, J. Gao, and D. Chen, “Research on vehicle routing problem with time windows based on improved Genetic Algorithm and Ant Colony Algorithm,” *Electronics (Basel)*, vol. 14, no. 4, p. 647, Feb. 2025, doi: 10.3390/electronics14040647.

[5] G. G. S. Putra, W. Swastika, and P. L. T. Irawan, “Perbandingan Particle Swarm Optimization dengan Genetic Algorithm dalam feature selection untuk analisis sentimen pada Permendikbudristek PPKS-LPT,” *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 8, no. 3, p. 412, Dec. 2022, doi: 10.26418/jp.v8i3.57300.

[6] I. Robandi, *Artificial Intelligence: mengupas rekayasa kecerdasan tiruan*. Penerbit Andi, 2021.

[7] J. Juwairiah, D. Pratama, H. C. Rustamaji, H. Sofyan, and D. B. Prasetyo, “Genetic Algorithm for optimizing traveling salesman problems with time windows (TSP-TW),” *International Journal of Artificial Intelligence & Robotics (IJAIR)*, vol. 1, no. 1, pp. 1–8, Nov. 2019, doi: 10.25139/ijair.v1i1.2024.

[8] J. Ochelska-Mierzejewska, “A Comparison of Ant Colony Optimization and Genetic Algorithm for solving the traveling salesman problem,” Wydawnictwo Politechniki Łódzkiej, vol. 24, no. 1, pp. 51–66, Mar. 2020, doi: 10.34658/jacs.2016.24.1.51-66.

[9] K. Buulolo and K. Wau, “Performance comparison of metaheuristic optimization algorithms in solving production scheduling problems,” Jurnal ICT : Information and Communication Technologies, 15(2), 122–132, doi: 10.35335/jict.v15i2.197.

[10] K. R. Harrison *et al.*, “A hybrid multi-population approach to the project portfolio selection and scheduling problem for future force design,” *IEEE Access*, vol. 9, pp. 83410–83430, 2021, doi: 10.1109/ACCESS.2021.3086070.

[11] M. Alhanjouri and B. Alfarra, “Ant Colony versus Genetic Algorithm based on traveling salesman problem,” International Journal Comp.Tech.Appl., vol. 2, no. 3, pp.570-78, 2012.

[12] M. G. Latheef, “Solving symmetrical traveling salesman problem,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. SP7, pp. 2629–2635, Jul. 2020, doi: 10.5373/JARDCS/V12SP7/20202399.

[13] R. Naufal and M. S. Hasibuan, “Optimization of distribution routes using the Genetic Algorithm in the traveling salesman problem,” *Journal of Applied Informatics and Computing*, vol. 9, no. 1, pp. 211–220, Jan. 2025, doi: 10.30871/jaic.v9i1.8864.

[14] V. Atiyatna, “Penyelesaian traveling salesman problem (TSP) asimetris dengan algoritma genetik commonality,” Universitas Jember, Jember, 2013.