# Hybrid Henry Gas Solubility Optimization: An Effective Algorithm for Fuel Consumption Vehicle Routing Problem

*by* Muhammad Faisal Ibrahim

**Submission date:** 12-Nov-2023 05:23AM (UTC+0700)
**Submission ID:** 2203047551
**File name:** 1_Full_Paper_Hybrid_He.pdf (285.27K)
**Word count:** 6245
**Character count:** 31347

**JURNAL ILMIAH TEKNIK INDUSTRI**
ISSN: 1412-6869 (Print), ISSN: 2460-4038 (Online)
Journal homepage: http://journals.ums.ac.id/index.php/jiti/index
doi: 10.23917/jiti.v20i2.15640

# Hybrid Henry Gas Solubility Optimization: An Effective Algorithm for Fuel Consumption Vehicle Routing Problem

Dana Marsetiya Utama [1a◆], Baiq Nurul Izzah Farida[1b], Ulfa Fitriani[1c], M. Faisal Ibrahim[2d], Dian Setiya Widodo[3e]

***Abstract.*** *The depletion of non-renewable fuel reserves is the biggest problem in the logistics sector. This problem encourages the transportation sector to increase fuel efficiency in distribution activities. The fuel optimization problem in distribution routing problems is called the Fuel Consumption Vehicle Routing Problem (FCVRP). This study proposes a novel Hybrid Henry Gas Solubility Optimization (HHGSO) to solve FCVRP problems. Experiments with several parameter variants were carried out to determine the performance of HHGSO in optimizing fuel consumption. Two data instances with 12 and 22 nodes were used to test the algorithm's effectiveness. In addition, the variation of population parameters, iterations HHGSO and the variable Kilometers per liter (KPL) are applied to optimize fuel consumption. The results show that the parameters of the HHGSO algorithm affect fuel consumption and computation time. In addition, the higher the KPL, the smaller the resulting fuel consumption. The proposed algorithm is also compared with several algorithms. The comparison results show that the proposed algorithm produces better computational time and fuel consumption than the Hybrid Particle Swarm Optimization, simulated annealing, and Tabu Search algorithms.*

***Keywords:*** *fuel consumption, algorithm Henry Gas, Vehicle Routing Problem, distribution*

## I. INTRODUCTION

Logistics is an essential element connecting companies to end customers (Garside et al., 2016). The harmful effects of greenhouse gas (GHG) on human health and the environment have become a serious topic in recent years (Pasaoglu et al., 2012). One sector that triggers this problem is the logistics sector (Ibrahim et al., 2020; Utama et al., 2021). Recently, researchers proposed green logistics to overcome this problem (Helo & Ala-Harja, 2018; Bi et al., 2020). Green logistics is directly related to production and distribution globally by considering environmental factors (Sbihi & Eglese, 2007). The transportation sector in distribution activities is the most significant contributor to non-renewable fuel consumption (Özener & Özkan, 2020). Most of the modes of transportation are conventional vehicles with non-renewable fuel consumption. Therefore, the transportation sector makes it a significant contributor to GHG in the atmosphere (Salimifard et al., 2012). Therefore, distribution activities are required to implement effective policies to minimize fuel consumption (Utama et al., 2021) to reduce the harmful effects of GHG (Abdoli et al., 2017). Vehicle Routing Problem (VRP) is a combinatorial problem in solving optimization problems in distribution routing activities (Poonthalir & Nadarajan, 2018; Utama et al., 2020; Ibrahim et al., 2021). Therefore, this problem is categorized as a combinatorial problem (Xiao & Konak, 2015; Ibrahim et al., 2021). The Green Vehicle Routing Problem (GVRP) recently introduced a new variant of VRP to help define routing policies at distribution centers. Lin et al. (2014) discussed the issue of GVRP, which focuses on optimizing fuel consumption. This study is commonly referred to as the fuel consumption vehicle routing problem (FCVRP)

[1] Industrial Engineering Department, Faculty of Engineering, Universitas Muhammadiyah Malang, Jalan Raya Tlogomas No. 246, Malang, 65144
[2] Logistics Engineering Department, Universitas Internasional Semen Indonesia, Jl. Veteran, Sidomoro, Kebomas, Gresik Regency, 61122
[3] Manufacturing Engineering, Vocational Faculty, Universitas 17 Agustus Surabaya, Jl. Semolowaru no. 45 Surabaya

[a] email: dana@umm.ac.id
[b] email: faridaabq@gmail.com
[c] email: ulfafitriani1@gmail.com
[d] email: faisalibrahim.ie@gmail.com
[e] email: diansetiyawidodo@untag-sby.ac.id
◆ corresponding author

Submited: 30-08-2021 Revised: 04-12-2021
Accepted: 06-12-2021

(Moghdani et al., 2020; Psychas et al., 2014; Rao et al., 2016).

Several studies to optimize fuel consumption have recently received more attention from researchers (Abdoli et al., 2017). The study focusing on FCVRP objectives was first investigated by Suzuki (2011). The proposed solution algorithm is the Tabu Search (TS) algorithm. Several other methods have also been proposed to minimize fuel consumption, including novel hybrid tabu search (Niu et al., 2018), intelligence heuristic partitioning (Gaur et al., 2013), and firefly algorithm (Zhang et al., 2015). Zhang et al. (2015) proposed a tabu search (TS) to solve the FCVRP problem. The Simulated annealing (SA) algorithm offered by Kuo (2010), which is also used by Normasari et al., (2019), Wang et al., (2019), and Xiao et al., (2012). Other popular algorithms that have been proposed for FCVRP problems include the ant colony optimization (ACO) algorithm, genetic algorithm (GA) , and particle swarm optimization (PSO) (Ramadhani & Garside, 2021). Ali and Farida (2021) recently proposed a Hybrid PSO (HPSO) procedure to solve the FCVRP problem. Unfortunately, their research does not consider computational time as a performance factor in solving FCVRP problems. In the decision-making, decision-makers are also considering procedures capable of completing fast FCVRP times with minimal energy consumption. However, previous studies did not analyze the computational time of FCVRP completion as one of the algorithm's performances.

One of the interesting new algorithms to be developed is the Henry Gas Solubility Optimization (HGSO) algorithm. This algorithm was proposed by Hashim et al. (2019), which was inspired by Henry law's physics principles. HGSO is proven can to solve optimization problems such as predicting tensile strength (Shehabeldeen et al., 2020), design optimization (Yıldız et al., 2020), feature selection (Neggaz et al., 2020), scheduling (Abd Elaziz & Attiya, 2021), controller design (Ekinci et al., 2020), prediction (Ding et al., 2021), and control optimization (Ekinci, et al., 2021). Although interest in FCVRP issues is increasing, studies related to FCVRP are still lacking. In addition, based on previous research, the HGSO Algorithm has never been applied to solving FCVRP. Nevertheless, it is what motivates researchers to use HGSO to complete FCVRP. This study tries to develop the HGSO procedure by combining neighborhood procedures such as flip, swap, and slide. Therefore, this study proposes a novel hybrid HGSO (HHGSO) procedure for optimizing the FCVRP problem. This study experimented with several parameters of the HHGSO algorithm to determine the performance of fuel consumption and computation time of completion. The proposed algorithm is also compared with several other algorithms to measure FCVRP performance and computation time. The contribution of this research is to enrich the procedure in the completion of FCVRP by proposing a novel procedure for HHGSO.

## II. Research Method

**Assumptions, Notations, and Mathematical Models**

In this section, this article describes the assumptions, notations, and mathematical models of FCVRP problems. The assumptions used are as follows; (1) Each vehicle departs and returns to the depot; (2) Each customer is visited by one vehicle; (3) Vehicles for logistics activities are homogeneous; (4) Fuel consumption is affected by distance, and weight of the load; and (5) The demand for each customer is constant.

The notation used in the FCVRP problem is as follows:

| | | |
|---|---|---|
| $P$ | : | the fuel increase % when increased vehicle load weight M |
| $N_r$ | : | Route/vehicle number |
| $J_r^i$ | : | the i-th node's on the r-th route (for example $J_1^2$= 1, the 2 path is 0-3-1-7-0, a node in the 2 is 3) |
| $Q$ | : | Vehicle capacity |
| $FC_{(J_r^i)(J_r^{i+1})}$ | : | fuel consumption from node i to i+1 on r-th route. |
| $L_{(J_r^i)(J_r^{i+1})}$ | : | vehicle load between node I to I+1 along the r-th route |
| $d_{(J_r^i)(J_r^{i+1})}$ | : | The distance in r-ths between nodes i-th and i+1 |
| $KPL_{(J_r^i)(J_r^{i+1})}$ | : | The distance traveled per liter fuel |

from i-th to i+1 node on the r th route

$v_{(R_r^i)(R_r^{i+1})}$ : the average speed of unloaded vehicles

M : When extra load weight M, the vehicle increases the fuel consumption

$V_r$ : Number of route nodes r, r = 1,2, ⋯, N

$q_{R_r^i}$ : Demand on the r-th route of the i-th node

The mathematical formulation of this study was developed from the FCVRP model proposed by Kuo (2010). The FCVRP mathematical formula for this problem is presented as follows:

$$\text{Min} \sum_{r=1}^{N_r} \sum_{s=1}^{v_r-1} \frac{d_{(R_r^i)(R_r^{i+1})}}{KPL_{(R_r^i)(R_r^{i+1})}} \times \left(1 + p \times \frac{L_{(R_r^i)(R_r^{i+1})}}{M}\right) \quad (1)$$

subject to

$$\sum_{i=2}^{V_r-1} q_{R_r^i} \leq Q, \qquad \forall r = 1, 2, \cdots, N_r \quad (2)$$

$$L_{(R_r^i)(R_r^{i+1})} = \sum_{i=i+1}^{V_r-1} q_{R_r^i}, \qquad \forall r = 1, 2, \cdots, N_r \quad (3)$$

$$R_r^1 = R_r^{V_r} = 0, \qquad \forall r = 1, 2, \cdots, N_r \quad (4)$$

$$N_r \geq 0, V_r \geq 0, R_r^i \subset V, \forall\, r = 1, 2, \cdots, N_r, \forall i = 1, 2, \cdots, V_r \quad (5)$$

The objective function of this problem is to minimize the cost of fuel consumption (Eq. (1)). The constraints of the FCVRP problem are modeled in 4 constraint equations. First, constraint to center as each vehicle route's starting and ensure that the load does not exceed the vehicle's capacity is formulated in Equations (2) and (3). Next, constraints that model the distribution ending nodes are formulated in Equation (4). The final constraint is to ensure that the number of vehicles and routes is > 0.

**Proposed Algorithm**

This study proposes the HHGSO algorithm inspired by the HGSO algorithm combined with the neighborhood procedure. HGSO is an algorithm initiated by Hashim et al. (2019) based on Henry law's physics principles. William Henry formulated Henry's law in 1803. Henry's law says that the solubility of a gas ($S_g$) is proportional to the partial pressure of the gas ($P_g$). It is formulated in equation (6), where H is the value of Henry's constant.

$$S_g = H \times P_g \quad (6)$$

HGSO has two main stages (Hashim et al., 2019) , such as (1) Initialization process and Clustering, and (2) update position including, Update Henry's Coefficient, Update Solubility, Escape from local optimum, and Update position of the worst agents. Furthermore, the HGSO algorithm is combined with the neighboring procedure in the proposed algorithm. Thus, the three main stages of the HHGSO algorithm are offered, including (1) Initialization process and Clustering (2) Position update includes, Update Henry's Coefficient, Update Solubility, Escape from local optimum, and Update position of the worst agents, and (3) Neighbor exchange. The complete stages of the proposed algorithm are presented in the following subsection.

*1) Initialization Process dan Clustering*

In this stage, the gas population (N) and the gas position vector are initialized according to equation (7). The position of the gas i in the population is denoted by $X_i$. $r$ is a random number 0 to 1. $X_{max}$ and $X_{min}$ indicate the upper bound and lower bound of the problem. Iteration is indicated by $t$. Henry's constant value $H_j(t)$, partial pressure of gas $i$ in cluster $j$ $(P_{i,j})$, and $\nabla_{sol}E/R$ constant value of $j(C_j)$ are modeled in equation (8). Where $l_1, l_2, l_3$ are constants with values $5E-02, 100$ and $1E-02$, respectively.

$$X_i(t+1) = X_{min} + r \times (X_{max} - X_{min}) \quad (7)$$

$$H_j(t) = l_i \times rand(0,1), P_{i,j} = l_2 \times rand(0,1), C_j = l_3 \times rand(0,1) \quad (8)$$

In the position initialization stage, each vector value for each gas is not allowed to have repeated values. The illustration of the initialization of the gas position vector can be seen in Figure 1. FCVRP is one of the combinatorial problems. Therefore, the HGSO position vector needs to be converted to discrete

space. This study proposes a Large Rank Value (LRV) procedure to transform the gas position vector to the travel sequence. LRV is easy to implement because the travel sequence is based on the largest position vector to the minor position vector (Utama, Widodo, Ibrahim, & Dewi, 2020a) (Utama, Baroto, & Widodo, 2020) (Utama & Widodo, 2021). Figure 2 is an illustration of converting gas position vectors to travel sequences with LRV.

$$P = \begin{Bmatrix} 0.54 & 0.11 & 0.61 \\ 0.29 & 0.65 & 0.71 \\ 0.38 & 0.43 & 0.99 \end{Bmatrix} \quad (a)$$

$$P = \begin{Bmatrix} 0.54 & 0.61 & 0.61 \\ 0.29 & 0.71 & 0.71 \\ 0.29 & 0.99 & 0.99 \end{Bmatrix} \quad (b)$$

**Figure 1.** Initialization of gas position vector (a) accepted gas position vector (b) rejected gas position vector

In this stage, the agent population (gas) is divided into several groups, equivalent to the type of gas amount. Each cluster has the same gas, so the value of the Henry constant $H_j$ is the same. Each $j$ cluster is evaluated to identify the best gas that reaches the best equilibrium condition from other gases. Furthermore, the gases are sequenced to obtain the optimal gas from the entire gas swarm.

*2) Update Position*

The gas position update is carried out every iteration ($t$). Henry's coefficient is updated each iteration based on equation (9). Where H_j is the Henry coefficient for cluster $j$. T indicates the temperature. $T^{\theta}$ is a constant with a value of 298.15, and the total number of iterations is denoted by $iter$.

$$H_j(t+1) = H_j(t) \times \exp\left(\left(-C_j \times (1/T(t) - 1/T^{\theta})\right)\right), \; T(t) = \exp(-t/iter) \quad (9)$$

The solubility of the gas is also updated using equation (10). The partial pressure of gas $i$ in cluster $j$ is denoted $P_{i,j}$, and $K$ is a constant. $S_{i,j}$ shows the solubility of gas $i$ in cluster $j$.

$$S_{i,j}(t) = K \times H_j(t+1) \times P_{i,j}(t) \quad (10)$$

The gas position vector update is formulated in equation (11). Where, $X_{(i,j)}$ describes the position of gas $i$ in cluster $j$. $r$ denotes a random value from 0 to 1, and $t$ is the $t$-th iteration. The best gas $i$ in cluster $j$ is denoted $X_{(i,best)}$, and the best gas in the swarm is formulated with $X_{best}$. In addition, the ability of gas $i$ in cluster $j$ to interact with other gases is formulated as $\gamma$. $\alpha$ shows the effect of other gases on gas $i$ in cluster $j$, which has a value of 1 and is a constant. $F$ is a flag that changes the direction of the search agent. The fitness value of gas $i$ in cluster $j$ is formulated as $F_{i,j}$. $F_{best}$ is the best gas fitness of the gas. The best gas $i$ in cluster $j$ is denoted as $X_{(i,best)}$. $X_{best}$ is the best gas in the swarm.

$$X_{i,j}(t+1) = X_{i,j}(t) + F \times rx\gamma \times \left(X_{i,best}(t) - X_{i,j}(t)\right) + F \times r \times \alpha \times \left(S_{i,j}(t) \times X_{best}(t) - X_{i,j}(t)\right)$$
$$\gamma = \beta \times exp\left(-\frac{F_{best}(t)+\varepsilon}{F_{i,j}(t)+\varepsilon}\right), \varepsilon = 0{,}05 \quad (11)$$

The escape from the local optimum procedure is performed in each iteration to avoid the local optimum solution. This procedure is modeled in equation (12). This procedure sorts the worst agents ($N_w$). Thus, N is the number of search agents. Furthermore, the next step is to update the position of the worst agents. This stage is formulated in equation (13). The position of gas $i$ in cluster $j$ is denoted as $G_{(i,j)}$. $r$ is a random number between 0-1. $G_{max}$ and $G_{min}$ show the upper bound and lower bound of the problem.

$$N_w = N \times (rand(c_2 - c_1) + c_1), c_1 = 0{,}1 \text{ dan } c_2 = 0{,}2 \quad (12)$$

$$G_{(i,j)} = G_{min(i,j)} + r \times \left(G_{max(i,j)} - G_{min(i,j)}\right) \quad (13)$$

*3) Neighborhood Exchange*

In this section, HGSO performance is improved with Neighborhood Exchange rules. HGSO archives the temporary solution and the best gas position vector for each iteration. Three rules of Neighborhood Exchange were adopted, namely, slide, flip, and swap, which are used to increase the solution performance at iteration t in each iteration. The slide is a Neighborhood Exchange rule by changing the position of the gas vector. One position vector for the gas is chosen randomly, then randomly transferred to one

vector. The illustration of this rule is shown in Figure 3. The flip rule is made by inverting the gas position vector. Two gas position vectors are selected randomly, and the vector is then reversed. This rule is illustrated in Figure 4. The Swap Rule is a rule for exchanging gas position vectors for two position vectors. Two gas position vectors are randomly selected to be replaced. This illustration is shown in Figure 5. This study suggests that each neighborhood exchange rule is repeated in each iteration 0.1 x the number of customers. Each neighborhood exchange is compared with the previous solution. The best solution is stored as the best solution in iteration $t$. The complete pseudo-code of the HHGSO algorithm is presented in Algorithm 1.

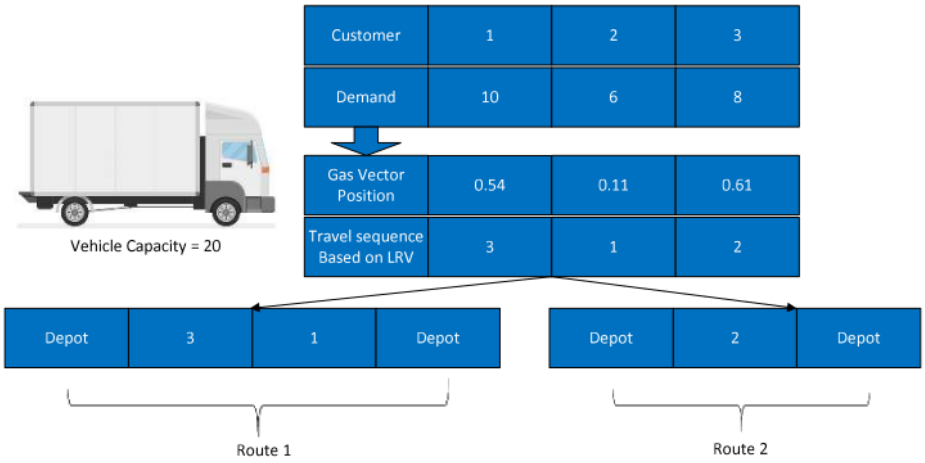


**Figure 2.** Illustration of converting gas position vector to travel sequence with LRV

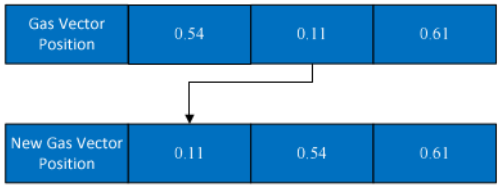


**Figure 3.** Illustration of Slide rules

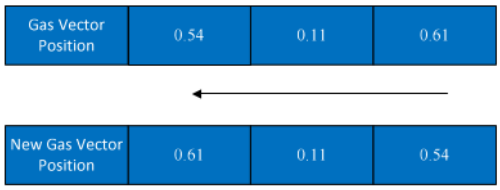


**Figure 4.** Illustration of the Flip rule

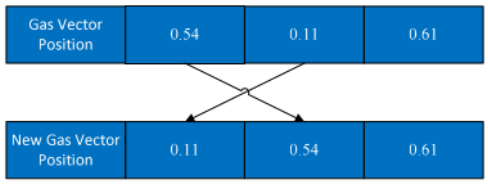


**Figure 5.** Illustration of swap rules

Algorithm 1 Pseudo-code algorithm HHGSO

Initialization: $X_i(1 = 1,2,..N)$, number of gas types I, $H_j, P_{i,j}, C_j, l_1, l_2$ and $l_3$
Sort the position using the Large Rank Value (LRV) method to get the vehicle route
Convert the vehicle route into sub route
Divide the population agents into some gas types (cluster) with the same henry's constant value ($H_j$)
Evaluate each cluster $j$
Get the best gas $X_{i,best}$ in each cluster, and the best search agent $X_{best}$
while $t <$ maximum number of iteration do
  for each search agent do
    update the positions of all search agents using Eq (11)
  end for
  Update Henry's coefficient of each gas type using eq (9)
  Update solubility of each gas using eq (10)
  Rank and select the number of worst agent using Eq (12)
  Update the position of the worst agents using Eq (13)
  Update the best gas $X_{i,best}$ and the best search agent $X_{best}$
for i = 0: 0.1 × n
  Perform slide on $X_{i,best}$ position
  if (evaluate ($X_{i,best}$) < evaluate ($X_{best}$))
  $X_{best} = X_{i,best}$
  end if
end for
for i = 0: 0.1 × n
  Perform flip on $X_{i,best}$ position
  if (evaluate ($X_{i,best}$) < evaluate ($X_{best}$))
    $X_{best} = X_{i,best}$
  end if
end for
for i = 0: 0.1 × n
  Perform swap on $X_{i,best}$ position
  if (evaluate ($X_{i,best}$) < evaluate ($X_{best}$))
    $X_{best} = X_{i,best}$
  end if
end for

end while
$t = t + 1$
Return $X_{best}$

## Data and Experimental

*Data*

This study used data instances presented in previous studies. In addition, data on the number of nodes, demand, distances were taken from the research of Dantzig and Ramser (1959) and Gaskell (1967). Node Numbers and Vehicle Capacity as shown in Table 1. The increase in fuel consumption ($\rho$) for each additional 45.35-kilogram load ($M$) was 2%.

In 12 node instances, distance and demands between nodes are shown table 2. Table 3 describes Coordinate and demands at 22 nodes. It is converted to be distance using equation (14).

$$d_{(J_r^i)(J_r^{i+1})} = \sqrt{(X_i - X_{i+1})^2 + (Y_i - Y_{i+1})^2} \quad (14)$$

**Table 1.** Node Numbers and Vehicle Capacity

| Number of Nodes | Vehicle Capacity | Sources |
|---|---|---|
| 12 | 6000 | Dantzig and Ramser (1959) |
| 22 | 4500 | Gaskell (1967) |

**Table 2.** Distance and demands between nodes at 12 node instances

| Node | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Demand |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 9 | 14 | 21 | 23 | 22 | 25 | 32 | 36 | 38 | 42 | 50 | 52 | 1200 |
| 1 | 9 | 0 | 5 | 12 | 22 | 21 | 24 | 31 | 35 | 37 | 41 | 49 | 51 | 1700 |
| 2 | 14 | 5 | 0 | 7 | 17 | 16 | 23 | 26 | 30 | 36 | 36 | 44 | 51 | 1500 |
| 3 | 21 | 12 | 7 | 0 | 10 | 21 | 30 | 27 | 37 | 43 | 31 | 37 | 39 | 1400 |
| 4 | 23 | 22 | 17 | 10 | 0 | 19 | 28 | 25 | 35 | 41 | 29 | 31 | 29 | 1700 |
| 5 | 22 | 21 | 16 | 21 | 19 | 0 | 9 | 10 | 16 | 22 | 20 | 28 | 30 | 1400 |
| 6 | 25 | 24 | 23 | 30 | 28 | 9 | 0 | 7 | 11 | 13 | 17 | 25 | 27 | 1200 |
| 7 | 32 | 31 | 26 | 27 | 25 | 10 | 7 | 0 | 10 | 16 | 10 | 18 | 20 | 1200 |
| 8 | 36 | 35 | 30 | 37 | 35 | 16 | 11 | 10 | 0 | 6 | 6 | 14 | 16 | 1900 |
| 9 | 38 | 37 | 36 | 43 | 41 | 22 | 13 | 16 | 6 | 0 | 12 | 12 | 20 | 1800 |
| 10 | 42 | 41 | 36 | 31 | 29 | 20 | 17 | 10 | 6 | 12 | 0 | 8 | 10 | 1600 |
| 11 | 50 | 49 | 44 | 37 | 31 | 28 | 25 | 18 | 14 | 12 | 8 | 0 | 10 | 1700 |
| 12 | 52 | 51 | 51 | 39 | 29 | 30 | 27 | 20 | 16 | 20 | 10 | 10 | 0 | 1100 |

**Table 3.** Coordinat and demands at 22 node instances

| Node | X | Y | Demand |
|---|---|---|---|
| 0 | 266 | 235 | 0 |
| 1 | 295 | 272 | 125 |
| 2 | 301 | 258 | 84 |
| 3 | 309 | 260 | 60 |
| 4 | 217 | 274 | 500 |
| 5 | 218 | 278 | 300 |
| 6 | 282 | 267 | 175 |
| 7 | 242 | 249 | 350 |
| 8 | 230 | 262 | 150 |
| 9 | 249 | 268 | 1100 |
| 10 | 256 | 267 | 4100 |
| 11 | 265 | 257 | 225 |
| 12 | 267 | 242 | 300 |
| 13 | 259 | 265 | 250 |
| 14 | 315 | 233 | 500 |
| 15 | 329 | 252 | 150 |
| 16 | 318 | 252 | 100 |
| 17 | 329 | 224 | 250 |
| 18 | 267 | 213 | 120 |
| 19 | 275 | 192 | 600 |
| 20 | 303 | 201 | 500 |
| 21 | 208 | 217 | 175 |
| 22 | 326 | 181 | 75 |

**Table 4** Fuel Consumption on case 12 node

| Iteration | KPL 9.35 | | | KPL 12.8 | | | KPL 16.25 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Gas Number | | | Gas Number | | | Gas Number | | |
| | 30 | 40 | 50 | 30 | 40 | 50 | 30 | 40 | 50 |
| 100 | 48.712 | 48.712 | 48.712 | 35.583 | 35.583 | 35.582 | 28.028 | 28.028 | 27.476 |
| 200 | 48.712 | 45.521 | **43.476** | 35.583 | 35.582 | **31.595** | 28.028 | 28.028 | **25.015** |
| 300 | 45.521 | 43.476 | **43.476** | 35.583 | **31.595** | **31.595** | 28.028 | **25.015** | **25.015** |

Bold shows the smallest fuel consumption

**Table 5** Fuel Consumption on case 22 node

| Iteration | KPL 9.35 | | | KPL 12.8 | | | KPL 16.25 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Gas Number | | | Gas Number | | | Gas Number | | |
| | 30 | 40 | 50 | 30 | 40 | 50 | 30 | 40 | 50 |
| 100 | 135.754 | 135.754 | 135.754 | 99.163 | 99.163 | 99.163 | 78.110 | 78.110 | 78.110 |
| 200 | 100.952 | 101.919 | 101.919 | 99.163 | 99.163 | 99.163 | 78.110 | 78.110 | 53.741 |
| 300 | 101.919 | 101.919 | **92.410** | 99.163 | 74.560 | **73.248** | 78.110 | 53.741 | **48.332** |

Bold shows the smallest fuel consumption

**Table 6** Computation time for 12 nodes

| Iteration | KPL 9.35 | | | KPL 12.8 | | | KPL 16.25 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Gas Number | | | Gas Number | | | Gas Number | | |
| | 30 | 40 | 50 | 30 | 40 | 50 | 30 | 40 | 50 |
| 100 | 5.732 | 6.775 | 8.579 | 5.280 | 6.189 | 8.791 | 5.270 | 8.664 | 8.954 |
| 200 | 10.843 | 13.752 | 16.896 | 11.043 | 13.596 | 16.388 | 10.734 | 13.567 | 16.439 |
| 300 | 15.317 | 19.486 | 25.354 | 15.960 | 19.196 | 24.105 | 15.997 | 19.129 | 25.924 |

**Table 7** Computation time for 22 nodes

| Iteration | KPL 9.35 | | | KPL 12.8 | | | KPL 16.25 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Gas Number | | | Gas Number | | | Gas Number | | |
| | 30 | 40 | 50 | 30 | 40 | 50 | 30 | 40 | 50 |
| 100 | 7.007 | 9.523 | 11.715 | 7.666 | 9.829 | 12.041 | 7.722 | 9.135 | 11.467 |
| 200 | 13.880 | 15.546 | 16.822 | 13.951 | 15.270 | 16.110 | 13.231 | 15.602 | 17.313 |
| 300 | 20.193 | 23.927 | 27.100 | 20.131 | 23.521 | 27.396 | 20.314 | 23.286 | 28.644 |

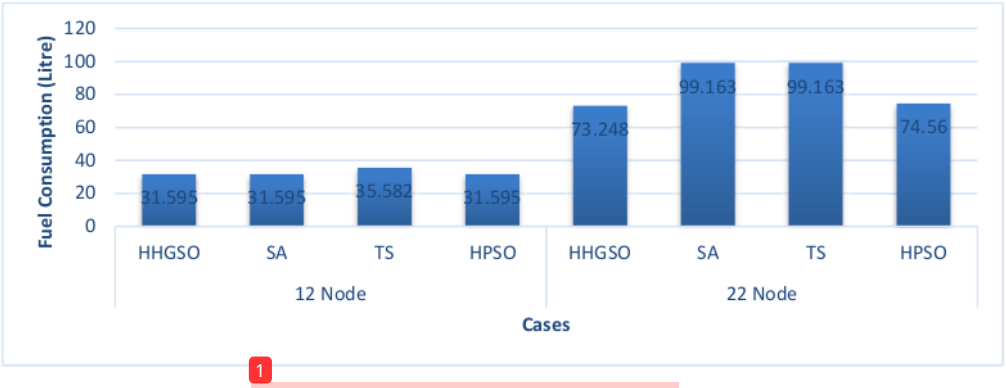


**Figure 6.** The results of the comparison of the algorithm to fuel consumption

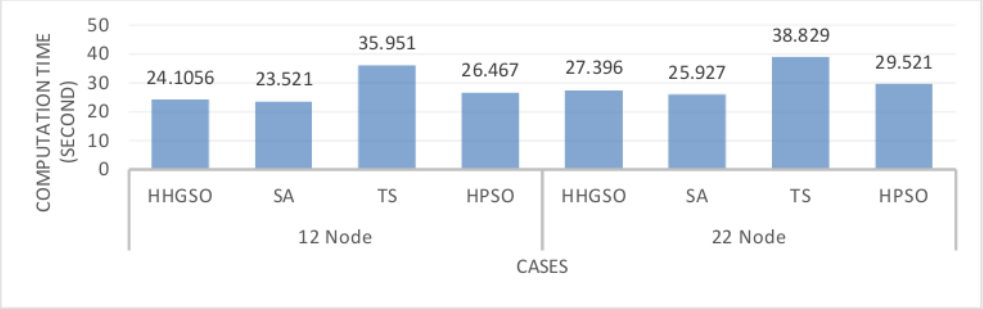


**Figure 7**. Comparison of algorithm results on computation time

**Table 8** REP results comparison algorithm

| Algorithm | REP 12 Node | REP 22 Node | Average |
|---|---|---|---|
| SA | 0.000 | 0.354 | 0.177 |
| TS | 0.126 | 0.354 | 0.240 |
| HPSO | 0.000 | 0.018 | 0.009 |

*Experimental*

The experiment was designed to determine changes in the number of gas, iteration, and KPL on fuel consumption and computation time. These have 3 data variations. The parameter number of gas used is 30, 40, and 50 gases. The iteration parameter utilized 100, 200, and 300 iterations. For KPL parameters, this study used 9.35, 12.8, and 16.25 kilometers per liter. In the HHGSO algorithm, this research employed 5 cluster numbers. The values of $M_1$ and $M_2$ were 0.1 and 0.2. In these cases, a total of 54 trials were applied.

The HHGSO algorithm was also compared with several algorithms such as SA (Kuo, 2010), TS (J. Zhang et al., 2015), and HPSO (Ali & Farida, 2021). The comparison used was in KPL 12.8. Each algorithm utilized 300 iterations. In the SA algorithm, this study employed the parameters population number = 50, Cooling rate = 0.8, and initial temperature = 100. In the Hybrid PSO algorithm, the Swarm size applied was 50, and the inertia weight applied was 0.6. Each experiment was run on Matlab 2014a software on windows 10 AMD A12 x64-64 8GB RAM processor.

Relative Error Percentage (REP) was used to measure the performance algorithm. It is formulated in equation (15). A high positive REP indicated that the proposed algorithm outperformed another algorithm. However, a negative REP showed that the proposed algorithm was not competitive compared to other algorithms.

$$\text{REP} = \frac{\text{FC}_{\text{Other\_algorithm}} - \text{FC}_{\text{proposed\_algorithm}}}{\text{FC}_{\text{proposed\_algorithm}}} \times 100\% \qquad (15)$$

## III. Result and Discussion

**Experimental results of HHGSO parameter changes**

This section presents the experimental results of HHGSO parameter changes on fuel consumption and computation time. The experimental results of changing HHGSO parameters on fuel consumption in 12 and 22 nodes are presented in Tables 4 and 5. Tables 6 and 7 describe the results of the computational time of experimental changes in HHGSO parameters. The experimental results show that the resulting fuel consumption becomes smaller when the gas number and iteration parameters increase. However, the computational time required is increasing. The results of this study are under the findings of Ali and Farida (2021), which states that iteration parameters and algorithm population affect fuel consumption.

In the experiment of changing the KPL on fuel consumption, the results show that when the KPL is increased, the resulting fuel consumption is smaller. Interestingly, this result is that when the KPL is changed at the same iteration and population, the resulting computational time is relatively the same. It shows that the KPL changes do not affect the same iteration and gas number computation time.

**Algorithm Comparison**

The results of the comparison of the algorithm to fuel consumption are shown in Figure 6. These results indicate that, in the case of 12 nodes, the proposed algorithm produces the same fuel consumption as SA and HPSO. However, the proposed algorithm is better than the TS algorithm. In the case of 22 Nodes, the proposed algorithm is better than HPSO, SA, and TS. Thus, It shows that the proposed algorithm effectively solves FCVRP problems.

The results of the REP algorithm comparison are presented in table 8. In the case of 12 nodes, the proposed algorithm produces the same solution as the SA and HPSO algorithms. It is evident from the REP value is 0. However, the REP value in the TS algorithm has a positive value. It indicates that the proposed algorithm is better than TS for the case of 12 nodes. In the case of 22 nodes, the REP SA, TS, and HPSO values produce values > 0. It shows that the proposed algorithm is better for the case of 22 nodes. The average REP results show that the proposed algorithm has better performance because the REP value of the proposed algorithm is more than 0.

The results of comparing algorithms on computational time are presented in Figure 7. These results show that the SA algorithm has the lowest computational time, followed by the HHGSO, HPSO, and TS algorithms. Although this SA algorithm produces the lowest computation time, the resulting fuel consumption is not optimal. The proposed HHGSO algorithm has better computational time than HPSO and TS. In addition, the computational time carried out by the proposed HHGSO algorithm is directly proportional to the resulting fuel consumption. Therefore, the HHGSO algorithm is effective for solving FCVRP problems.

## IV. Conclusion

This study addressed the FCVRP problem by proposing a new algorithm HHGSO. Experiments were carried out with various variations of iteration parameters and gas numbers. The results show that the gas number and iteration parameters affect fuel consumption and computation time. The proposed HHGSO algorithm produced lower fuel consumption than other algorithms based on the comparison algorithm. It is evident from the REP value of the comparison algorithm. In addition, the computational time generated by HHGSO can compete with the TS and HPSO algorithms. However, this research is limited to solving the problem of 1 distribution center. Future studies need to consider multi-distribution centers in the case of FCVRP.

## Unknowledgement

On this occasion we would like to thank the Universitas Muhammadiyah Sidoarjo for facilitating funding and other research facilities through the 2020 research institution grant.

## References

Abd Elaziz, M., & Attiya, I. (2021). An improved Henry gas solubility optimization algorithm for task scheduling in cloud computing. *Artificial Intelligence Review, 54* (5), 3599-3637.

Abdoli, B., MirHassani, S. A., & Hooshmand, F. (2017). Model and algorithm for bi-fuel vehicle routing problem to reduce GHG emissions. *Environmental Science and Pollution Research, 24* (27), 21610-21624.

Ali, M., & Farida, B. N. I. (2021). Completion of FCVRP using Hybrid Particle Swarm Optimization Algorithm. *Jurnal Teknik Industri, 22* (1), 98-112.

Bi, K., Yang, M., Zahid, L., & Zhou, X. (2020). A New Solution for City Distribution to Achieve Environmental Benefits within the Trend of Green Logistics: A Case Study in China. *Sustainability, 12* (20), 8312.

Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science, 6* (1), 80-91. doi: 10.1287/mnsc.6.1.80

Dewi, S. K., & Utama, D. M. (2021). A New Hybrid Whale Optimization Algorithm for Green Vehicle Routing Problem. *Systems Science & Control Engineering, 9* (1), 61-72. doi: 10.1080/21642583.2020.1863276

Ding, W., Nguyen, M. D., Mohammed, A. S., Armaghani, D. J., Hasanipanah, M., Van Bui, L., & Pham, B. T. (2021). A new development of ANFIS-Based Henry gas solubility optimization technique for prediction of soil shear strength. *Transportation Geotechnics, 29*, 100579.

Ekinci, S., Hekimoğlu, B., & Izci, D. (2021). Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor. *Engineering Science and Technology, an International Journal, 24* (2), 331-342.

Ekinci, S., Izci, D., & Hekimoğlu, B. (2020). *Henry Gas Solubility Optimization Algorithm Based FOPID Controller Design for Automatic Voltage Regulator*. Paper presented at the 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE).

Garside, A. K., Sulistyani, X., & Utama, D. M. (2016). *Penentuan rute distribusi LPG dengan pendekatan model matematis*. Paper presented at the Prosiding SENTRA (Seminar Teknologi Dan Rekayasa).

Gaskell, T. (1967). Bases for vehicle fleet scheduling. *Journal of the Operational Research Society, 18* (3), 281-295.

Gaur, D. R., Mudgal, A., & Singh, R. R. (2013). Routing vehicles to minimize fuel consumption. *Operations Research Letters, 41* (6), 576-580.

Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W., & Mirjalili, S. (2019). Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems, 101*, 646-667. doi: https://doi.org/10.1016/j.future.2019.07.015

Helo, P., & Ala-Harja, H. (2018). Green logistics in food distribution–a case study. *International Journal of Logistics Research and Applications, 21* (4), 464-479.

Ibrahim, M. F., Nurhakiki, F. R., Utama, D. M., & Rizaki, A. A. (2021). *Optimised Genetic Algorithm Crossover and Mutation Stage for Vehicle Routing Problem Pick-Up and Delivery with Time Windows*. IOP Conference Series: Materials Science and Engineering, 1071(1), 012025. doi: 10.1088/1757-899x/1071/1/012025

Ibrahim, M. F., Putri, M., Farista, D., & Utama, D. M. (2021). An Improved Genetic Algorithm for Vehicle Routing Problem Pick-up and Delivery with Time Windows. *Jurnal Teknik Industri, 22* (1), 1-17.

Ibrahim, M. F., Putri, M. M., & Utama, D. M. (2020). *A literature review on reducing carbon emission from supply chain system: drivers, barriers, performance indicators, and practices*. IOP Conference Series: Materials Science and Engineering, 722, 012034. doi: 10.1088/1757-899x/722/1/012034

Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering, 59* (1), 157-165.

Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., & Lam, H. (2014). Survey of green vehicle routing problem: past and future trends. *Expert Systems with Applications, 41* (4), 1118-1138.

Moghdani, R., Salimifard, K., Demir, E., & Benyettou, A. (2020). The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production*, 123691.

Neggaz, N., Houssein, E. H., & Hussain, K. (2020). An efficient henry gas solubility optimization for feature selection. *Expert Systems with Applications, 152*, 113364.

Niu, Y., Yang, Z., Chen, P., & Xiao, J. (2018). A hybrid tabu search algorithm for a real-world open vehicle routing problem involving fuel consumption constraints. *Complexity*, 2018.

Normasari, N. M. E., Yu, V. F., & Bachtiyar, C. (2019). A simulated annealing heuristic for the capacitated green vehicle routing problem. *Mathematical Problems in Engineering*, 2019.

Norouzi, N., Sadegh-Amalnick, M., & Tavakkoli-Moghaddam, R. (2017). Modified particle swarm optimization in a time-dependent vehicle routing problem: minimizing fuel consumption. *Optimization Letters, 11* (1), 121-134.

Özener, O., & Özkan, M. (2020). Fuel consumption and emission evaluation of a rapid bus transport system at different operating conditions. *Fuel, 265*, 117016.

Pasaoglu, G., Honselaar, M., & Thiel, C. (2012). Potential vehicle fleet CO2 reductions and cost implications for various vehicle technology deployment scenarios in Europe. *Energy Policy, 40*, 404-421.

Poonthalir, G., & Nadarajan, R. (2018). A fuel efficient green vehicle routing problem with varying speed constraint (F-GVRP). *Expert Systems with Applications, 100*, 131-144.

Psychas, I.-D., Marinaki, M., Marinakis, Y., & Migdalas, A. (2014). *Minimizing the fuel consumption of a multiobjective vehicle routing problem using the parallel multi-start NSGA II algorithm*. Paper presented at the International Conference on Network Analysis.

Ramadhani, B. N. I. F., & Garside, A. K. (2021). Particle Swarm Optimization Algorithm to Solve Vehicle Routing Problem with Fuel Consumption Minimization. *Jurnal Optimasi Sistem Industri, 20* (1), 1-10.

Rao, W., Liu, F., & Wang, S. (2016). An efficient two-objective hybrid local search algorithm for solving the fuel consumption vehicle routing problem. *Applied Computational Intelligence and Soft Computing*, 2016.

Salimifard, K., Shahbandarzadeh, H., & Raeesi, R. (2012). *Green transportation and the role of operation research*. Paper presented at the Int. Conf. Traffic Transp. Eng.(ICTTE 2012).

Sbihi, A., & Eglese, R. W. (2007). Combinatorial optimization and green logistics. *4OR, 5* (2), 99-116.

Shehabeldeen, T. A., Abd Elaziz, M., Elsheikh, A. H., Hassan, O. F., Yin, Y., Ji, X., & Zhou, J. (2020). A novel method for predicting tensile strength of friction stir welded AA6061 aluminium alloy joints based on hybrid random vector functional link and henry gas solubility optimization. *IEEE Access, 8*, 79896-79907.

Suzuki, Y. (2011). A new truck-routing approach for reducing fuel consumption and pollutants emission. *Transportation Research Part D: Transport and Environment, 16* (1), 73-77. doi: https://doi.org/10.1016/j.trd.2010.08.003

Utama, D. M., Baroto, T., & Widodo, D. S. (2020). Energy-Efficient Flow Shop Scheduling Using Hybrid Grasshopper Algorithm Optimization. *Jurnal Ilmiah Teknik Industri, 19* (1), 30-38.

Utama, D. M., Dewi, S. K., Wahid, A., & Santoso, I. (2020). The vehicle routing problem for perishable goods: A systematic review. *Cogent Engineering, 7* (1), 1816148. doi: 10.1080/23311916.2020.1816148

Utama, D. M., Fitria, T. A., & Garside, A. K. (2021). *Artificial Bee Colony Algorithm for Solving Green Vehicle Routing Problems with Time Windows*. Paper presented at the Journal of Physics: Conference Series.

Utama, D. M., & Widodo, D. S. (2021). An energy-efficient flow shop scheduling using hybrid Harris hawks optimization. *Bulletin of Electrical Engineering and Informatics, 10* (3), 1154-1163.

Utama, D. M., Widodo, D. S., Ibrahim, M. F., & Dewi, S. K. (2020a). An effective hybrid ant lion algorithm to minimize mean tardiness on permutation flow shop scheduling problem. *International Journal of Advances in Intelligent Informatics, 6* (1), 23-35.

Utama, D. M., Widodo, D. S., Ibrahim, M. F., & Dewi, S. K. (2020b). A New Hybrid Butterfly Optimization Algorithm for Green Vehicle Routing Problem. *Journal of Advanced Transportation*, 2020.

Wang, R., Zhou, J., Yi, X., & Pantelous, A. A. (2019). Solving the green-fuzzy vehicle routing problem using a revised hybrid intelligent algorithm. *Journal of Ambient Intelligence and Humanized Computing, 10* (1), 321-332. doi: 10.1007/s12652-018-0703-9

Xiao, Y., & Konak, A. (2015). A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness. *Applied Soft Computing, 34*, 372-388.

Xiao, Y., Zhao, Q., Kaku, I., & Xu, Y. (2012). Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & operations research, 39* (7), 1419-1431.

Xiong, H. (2010). *A fuel consumption objective of VRP and the genetic algorithm*. Paper presented at the 2010 International Conference on E-Product E-Service and E-Entertainment.

Yao, E., Lang, Z., Yang, Y., & Zhang, Y. (2015). Vehicle routing problem solution considering minimising fuel consumption. *IET Intelligent Transport Systems, 9* (5), 523-529.

Yıldız, B. S., Yıldız, A. R., Pholdee, N., Bureerat, S., Sait, S. M., & Patel, V. (2020). The Henry gas solubility optimization algorithm for optimum structural design of automobile brake components. *Materials Testing, 62* (3), 261-264.

Zhang, J., Zhao, Y., Xue, W., & Li, J. (2015). Vehicle routing problem with fuel consumption and carbon emission. *International Journal of Production Economics, 170*, 234-242.

Zhang, Z., Wei, L., & Lim, A. (2015). An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. *Transportation Research Part B: Methodological, 82*, 20-35. doi: https://doi.org/10.1016/j.trb.2015.10.001

# Hybrid Henry Gas Solubility Optimization: An Effective Algorithm for Fuel Consumption Vehicle Routing Problem

ORIGINALITY REPORT

| 5% | 3% | 6% | 0% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | www.beei.org<br>Internet Source | 3% |
| 2 | Shanty Kusuma Dewi, Dana Marsetiya Utama. "A New Hybrid Whale Optimization Algorithm for Green Vehicle Routing Problem", Systems Science & Control Engineering, 2020<br>Publication | 3% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 3% |
| Exclude bibliography | On | | |