Soft Computing
https://doi.org/10.1007/s00500-023-09574-y

SOFT COMPUTING IN DECISION MAKING AND IN MODELING IN ECONOMICS

Varied offspring memetic algorithm with three parents for a realistic synchronized goods delivery and service problem

Somnath Maji[1] · Samir Maity[2] · Sumanta Bsau[2] · Debasis Giri[3] · Manoranjan Maiti[4]

Accepted: 10 December 2023

Abstract

In a competitive online retail market, orders for assembled products such as refrigerators, air conditioners, smart televisions, etc., attract significant attention due to their high gross merchandise value. Unlike other regular products, product delivery has a two-stage process—delivery of product components and assembly and installation of the final product—involving multiple parties that may be internal or external to the organization. Coordination of the above activities is essential to reduce customer dissatisfaction and to curb the various waiting or demurrage costs due to delayed arrivals of goods vehicles and traveling salesman. This paper attempts to model and solve such a realistic synchronized goods delivery and service problem against the online booking. In this model, one goods vehicle starts from the company's storehouse with all the goods to be delivered and moves continually, dropping the goods at the specified locations. For service, a traveling salesman separately moves and uses the appropriate conveyances among the available ones at each node to reach the customers. This paper poses some interesting research questions to understand the requirements of separate tour paths for goods vehicles and traveling salesman along with appropriate conveyance for traveling salesman's arrival. This is an NP-hard traveling salesman problem. For solving, a varied offspring memetic algorithm (VOMA) with modified probabilistic selection, varied offspring three-parent (i.e., surro-embryos) crossover and Fibo-generation-dependent mutation is developed and tested on some standard test functions to establish its superiority over the standard ones. VOMA implementation on the above proposed problem reveals the influence of unloading and service times, halt time and third-party outsourcing charges on the final optimum route design. Finally, the paper provides a structured decision-making framework for practitioners and showcases a case study by implementing VOMA in a similar problem context.

Keywords Memetic algorithm (MA) · Modified probabilistic selection · Traveling salesman problem and servicing · Three-parent crossover

1 Introduction

A traveling salesman problem (TSP) aims to determine the round-trip tour for a traveling salesman/service person who visits a number of cities, each one exactly once, and returns to the starting city so that the total travel distance, cost and/or

✉ Samir Maity
samirm@iimcal.ac.in

Somnath Maji
somnathmajivucs@gmail.com

Sumanta Bsau
sumanta@iimcal.ac.in

Debasis Giri
debasis_giri@hotmail.com

Manoranjan Maiti
mmaiti2005@yahoo.co.in

1 Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Nadia 741249, WB, India

2 Operations Management Group, Indian Institute of Management Calcutta, D H Road, Joka, Kolkata 700104, India

3 Department of Information Technology, Maulana Abul Kalam Azad University of Technology, Nadia 741249, WB, India

4 Department of Applied Mathematics with Oceanology and Computer Programming, Vidyasagar University, Midnapur 721102, India

Published online: 17 January 2024

time are minimal. The TSP has applications in many real-life problems, such as the vehicle routing problem (VRP), the facility location problem (FLP), disaster relief systems, etc.

"We need to step faster into the digital innovations, where we can meet our customers digitally $\cdots$" said the CEO of IKEA, India (cf. Bailay 2018). IKEA is a Swedish furniture company with 355 stores in 29 countries, primarily engaged in selling ready-to-assemble furniture via a home delivery system, i.e., by sending furniture components to the customer locations as per online orders, with an assembler (i.e., the company's traveling salesman) to assemble the furniture. A similar delivery model is used for products like air conditioners, water purifiers (cf. Sarkar 2014), etc. Manufacturers or distributors supply goods against orders and send service people for installation. Xiaomi Corporation, an international Chinese electronics company, sells smart TVs on order through home delivery and later performs installation by sending own service worker (cf. Corporation 2012). We generalize these situations by relating them to an assemble-to-order scenario, in which the final product delivery is accomplished at the customer location involving multiple parties. Practical situations might demand that the parties belong to the same organization for better coordination or to different organizations for the utilization of trained human resources. We understand that synchronization of arrival times is necessary to avoid the penalty costs paid to customers or the cost of waiting by the trained personnel assigned for the final product delivery to the customer. With the objective of minimizing the total cost, a mathematical formulation is necessary to understand the optimization model of the underlying problem context. Thus, our first research contribution in this paper involves developing a mathematical model to define the time-synchronized multiple TSPs for optimal traveling salesman travel and goods delivery with service. In the following sections, we refer to this problem as 'Solid TSP with Goods Delivery and Services' (STSPwGDS) for brevity.

In this type of delivery model, the consignment begins its journey from a location (depot) by a particular vehicle that delivers required units to different locations/nodes and returns to the depot. The traveling salesman also starts at the same time from the depot to traverse those customer locations. In real-life situations, the products are dispatched from the company warehouse, and a third-party service personnel arrives at the warehouse to obtain the list of customers for installations following the delivery schedule. Thus, the travel routes for traveling salesman and the goods' vehicle might be the same or different depending on the vehicle type or service time. Our second research question is whether keeping the same route for both goods vehicle and traveling salesman will be appropriate or whether the routes might be changed because of the involvement of other cost components. In answering this question, we find the appropriate travel routes for both traveling salesman and goods vehicle so that the overall cost for the system is minimal.

We consider a single vehicle type for the movement of the goods vehicle while providing flexibility of multiple vehicle types for salesperson travel from one location to another. This model is an extension of the conventional 2D TSP and is defined as a 3D TSP because of adding one more dimension, i.e., vehicle type. Following this model, traveling salesman can change the vehicle type to traverse from one customer location to another based on cost, time, and availability. This choice is particularly relevant in developing countries, where expanding e-commerce models depends on geographic reach. Our third research question attempts to understand the implications of vehicle type in designing the routes for traveling salesman in STSPwGDS.

We have also studied two other variations of the models developed, in which we allow the goods vehicle to halt, not exceeding a certain time limit, to avoid the cost of delivering the goods early. We have also attempted to understand the impact on cost savings with an increase in the delivery time window (cf. Forum 2018). With our fourth research question, we attempt to understand the utility of this additional provision of allowing halting time in the route design and in the total cost, along with the sensitivity of decision-making to parameters from the problem context.

STSPwGDS is an NP-hard problem. Considering the NP-hard nature of the problem, our fifth research contribution is to develop a novel heuristic based on a genetic algorithm. Mimicking the real-life '3-parent'/'three-parent' childbirth system (cf. BBC 2018; Reardon 2017), a varied offspring memetic algorithm (VOMA) is developed as a modified GA with a modified probabilistic selection, varied offspring surro-embryos crossover and fibo-generation-dependent (Fibo-GD) mutations. The above proposed STSPwGDS and its variants (including the service provided by a third-party) are solved and numerically illustrated by the developed VOMA. Statistical tests are performed to conduct a robust comparison of the proposed VOMA on benchmark TSPLIB problems (cf. Reinelt 1995). The efficiency of VOMA is evinced statistically by Friedman test and (post hoc) paired comparisons.

Apart from addressing the aforementioned research questions, we establish the validity of our proposed algorithm by solving a real-life problem of a furniture delivery firm with a problem setup similar to the proposed model. We also analyze the impact of problem context parameters on total cost and decision-making by performing sensitivity analysis. We further extend the understanding by developing a generic decision-making framework across the problem areas.

The novelties in this investigation are as follows:

- This study formulates product delivery routing and service independently.
- The product components' delivery and installation are synchronized.
- The routing approach considers the influence of unloading, service, wasted time, and third party.
- A varied offspring memetic algorithm (VOMA) with modified probabilistic selection is proposed.
- The VOMA considers varied offspring three-parent crossover and fibo-generation-dependent mutation.

The paper is organized as follows. Section 1 presents a brief introduction of the problem context, and the heuristics developed, followed by Sect. 2, which provides a literature review. The mathematical description of the problem is elaborated in Sect. 3. Section 4 explains the VOMA heuristic, with Sect. 5 providing details about computational experiments to establish the efficiency of the VOMA through statistical tests. Section 6 conducts a performance analysis of the heuristic and provides a follow-up discussion. Practical implementation and the managerial insights obtained by implementing the heuristic are elaborated in Sects. 7 and 8, respectively. We conclude the paper by summarizing the major contributions and indicating future research avenues in Sect. 9.

2 Literature review

2.1 Different types of TSP

The TSP is a combinatorial NP-hard optimization problem (Lawler et al. 1985; Lee et al. 2022). Different researchers have studied several kinds of TSPs over the last few decades. Some of them are TSPs with time windows (Focacci et al. 2002), stochastic TSPs (Chang et al. 2009), double TSPs (Petersen and Madsen 2009), asymmetric TSPs (Majumdar and Bhunia 2011), constrained TSPs (Moon et al. 2002), etc.

2.2 Goods delivery and third party

Recently, variants of TSPs related to goods delivery and service have gained traction in scholarly articles because of their increasing relevance to e-commerce business models. Averbakh and Yu (2018) developed algorithms for multi-depot traveling problems against service calls generated by nodes of a transportation network independently with known probabilities. Carrabs et al. (2017) formulated a variant of the VRP for urban grocery delivery problem and solved it through a mixed-integer linear programming model. They also considered distance constraints, emissions, and street traffic limitations in urban areas satisfying the customer demand arrived through grocery e-channels. Feng (2019) proposed a third-party distribution model for fruit and vegetable agricultural products solved through ant colony algorithm. The main focus was on difficulties in the urban as well as rural areas of the third-party distribution procedure.

Malaguti et al. (2018) formulated a goods pickup and delivery problem in maritime logistics with a ship visiting different ports. In their paper, the authors developed heuristic procedures and a branch-and-cut approach. Cordeau et al. (2007) discussed the necessity and usefulness of transportation on demand (TOD) and presented some static and dynamic TOD problems as generalizations of the vehicle routine problems with pickup and delivery. Wang and Lin (2017) incorporated travel time uncertainty into the design of service regions for pickup and delivery problems with time windows. Although the time uncertainty part has been addressed in the published literature, asynchronous delivery with multiple TSPs has not been addressed in the published literature.

2.3 Multi-vehicle TSP

A multi-index transportation problem was first designed by Haley (1963) by adding types of vehicles as a decision variable. Initial work on 3D TSPs was published by Haxhimusa et al. (2011). Although some initial representations of 3D TSPs were illustrated by Haxhimusa et al. (2011), the authors did not focus on the practical significance of 3D TSPs to real-life problems. Maity et al. (2015) proposed a heuristic to solve a 3D TSP that considered vehicle type along with route design. Maity et al. (2017) and Roy et al. (2016) extended the above problem on restricted 3D TSPs and 4D TSPs by exploring the possibility of multiple paths along with vehicle type between two cites. We observe similar applications in maritime transportation (cf. Constantinescu 2012) with variations in vessel types and route design. To the best of our knowledge, there is no paper that explores the possibility of multiple vehicle types in time-synchronized TSPs.

2.4 Genetic algorithm for TSP

To solve these NP-hard combinatorial optimization problems (Beamurgia et al. 2022) within a reasonable time, heuristic methods such as genetic algorithm (GA), ant colony optimization (ACO), simulated annealing (SA), etc., are used. Within the heuristics used, GA received significant traction because of its performance in solution quality obtained and computational time. Traditional GA techniques are modified to yield better results by creating specific problem structures, such as making the underlying graph sparse by Wang (2015). Nagata and Soler (2012) extended GA using an edge assembly crossover operator to solve asymmetric TSPs; Dong et al. (2012) presented a new hybrid algorithm, the cooperative genetic ant system, combining both GA and ACO in a cooperative manner to solve TSPs. A novel mutation called the greedy sub-tour mutation was introduced with simple GA by Albayrak and Allahverdi (2011). Ma et al. (2019) studied a priority-based nested genetic algorithm, where they used weight mapping crossover, fuzzy logic-based adjusted mutation rate to solve a variant of VRP. Xu et al. (2019) proposed a GA with one-by-one revision of two sides which is an approximate algorithm to obtain optimal Hamiltonian circuit. They also focused on optimal goods distribution routes with real-time traffic information for delivery staff.

2.5 Multi-parent multi-offspring GA

Early works of multi-parent recombination mechanisms in GA include a paper by Eiben et al. (1994) using gene scanning and diagonal crossover. Recently, Rodriguez-Roman (2018) used a surrogate for the joint selection and design of highway safety and travel time improvement projects formulated as a bi-objective, mixed integer optimization problem with constraints. Wang et al. (2016) was the first study to introduce a multi-offspring GA (MO-GA) on TSPs, in accordance with biological, evolutionary, and mathematical ecological theory. Recently, Lagarteja et al. (2017) studied an improved GA using a new crossover operator called Path and Pob genes exchange operator (PPX) and compared its performance with six already available crossover operators. Our paper differs from the existing GA implementation by generating random numbers of offspring from a multi-parent crossover to enable diversification.

2.6 Memetic algorithm

Once introduced by Moscato et al. (1989), the "memetic algorithm (MA)" established its potential to provide better solutions for TSPs by combining local search or evolutionary algorithms with traditional GA. Some examples include hybrid evolutionary algorithms (cf. Martínez-Estudillo et al. 2005), Baldwinian evolutionary algorithms (cf. Baldwin 1896) and Lamarckian evolutionary algorithms (cf. Skinner 2015). Wang et al. (2010) proposed effective MAs to solve TSPs based on two improved Inver-over operators to increase the convergence speed. Merz and Freisleben (2001) focused on the fitness landscapes of several instances of the TSP. They used new generic recombination-based MAs, which exploited the correlation structure to identify near-optimal tours. Ghoseiri and Sarhadi (2008) introduced a specially designed MA to solve the symmetric TSPs, using a local search combined with a specially designed GA. Recently, Tüű-Szabó et al. (2017) modified MA, which is called the discrete bacterial memetic evolutionary algorithm, to solve the TSP with time windows. This method is the combination of the bacterial evolutionary algorithm with two-opt and three-opt local searches. Ye et al. (2014) presented a multi-parent MA to solve the classic linear ordering problem. The MPM algorithm integrates a multi-parent recombination operator to generate offspring solutions and a distance- and quality-based criterion for pool updating. Extending these MA strategies, we propose a novel MA implementation technique, considering the varied offspring multi-parent strategy.

2.7 Operators of GA

Majumdar and Bhunia (2011) introduced an elitism-based selection process. First, the current population is sorted from best to worst in terms of interval-valued fitness. By comparing interval numbers, a proportion of the better individuals are copied from the current generation to the next. Maity et al. (2015) introduced a probabilistic selection procedure to obtain better chromosomes for an optimal solution in a smaller number of generations. Moon et al. (2002) used a mixed strategy based on the roulette wheel, and elitist selection was adopted as the selection procedure by choosing chromosomes from the population space based on either parent and offspring or parts of them. Majumdar and Bhunia (2011) introduced the exchange and replacement mutation in GA to obtain a better solution for asymmetric TSPs. Albayrak and Allahverdi (2011) developed a new mutation operator (greedy sub tour mutation) to increase GA performance. In VOMA, we have developed novel modified probabilistic selection to obtain faster and better solutions and a fibo-generation-dependent mutation for the smooth generation of the probability of crossover.

3 Problem formulation

This section contains the TSP formulation, along with its variants. Section 3.3 presents standard TSP formulation. Section 3.4 extends it by considering multiple vehicle types. In Sect. 3.5, we formulate the proposed STSPwGDs problem by including goods delivery and service within TSPs.

3.1 Nomenclature

Table 1 presents the notations of some parameters and decision variables used in the problem. We assume the following assumptions in building up the models in this investigation.

3.2 Assumptions

(i) There are $(N-1)$ customers and one depot from which the goods vehicle and the traveling salesman start their journey.
(ii) Against online bookings, a wholesaler sends goods to the customers' locations (by car) for dropping.

Table 1 Notations and descriptions of parameters and decision variables

Notations	Descriptions
N	Number of nodes, $N = 1$ is the depot
i, j, k	Index sets
x_{ij}	Decision variable, when traveling from ith city to jth city $x_{ij} = 1$, else, $x_{ij} = 0$
$c(i, j)$	Traveling cost from ith city to jth city
$\alpha(x_i, x_{i+1}, v_p)$	Traveling cost for the traveling salesman from ith city to $(i + 1)$th city using (p)th vehicle per unit distance
$\beta(y_i, y_{i+1}, V)$	Traveling cost for the transport vehicle from ith city to $(i + 1)$th city using (V)th vehicle per unit distance
$h(x_j, x_{j+1}, v_p)$	Traveling time for the traveling salesman from jth city to $(j + 1)$th city using (p)th vehicle
$g(x_j, x_{j+1}, V)$	Traveling time for the transport vehicle from jth city to $(j + 1)$th city using (V)th vehicle
$(v_1, v_2, \ldots, v_q)$	Different vehicle, q is the number of available vehicles
$\text{dis}(x_i, x_{i+1})$	Distance between ith city to $(i + 1)$th city
$\text{dis}(x_N, x_1)$	Distance between Nth city to depot
ξ	Goods transportation charges per unit weight per unit distance
d_i	Demand at ith node (kg), d_1=0
D	Total demand (kg)
w_3	Servicing cost per unit weight
w_4	Unloading cost per unit weight
w_6	Third-party servicing cost per unit weight
t_1	Servicing time per unit weight
t_2	Unloading time per unit weight
η_1	Waiting cost of traveling salesman per unit time
η_2	Halting cost of transport vehicle per unit time
η_4	Waiting cost of traveling salesman per unit time
η_5	Fixed charge for every halt
$\Pi(i)$	Penalty cost of the system
$\Pi^s(i)$	Waiting time of the traveling salesman at ith node
Ω_2	Daytime halt limit
θ_i	Entry time of the service vehicle at (i)th node
ϕ_i	Exit time of the transport vehicle at (i)th node
t^{ct}	Current time
tth	Starting time
t^{tl}	Time limit
s'	Maximum number of offspring
TNOP	Total number of parents
TNC	Total number of crossover
RON	Random offspring number
$-i(a) - j(b)-$	Travel from node (i) to node (j) through vehicle a, $a = 0, 1, 2$

All costs and times are in \$ and hour, respectively, and time is measured in 24-h format

(iii) Three types of installation scenarios are considered. The goods vehicle and traveling salesman (service man) start the journey at the same time, but (a) they move separately, (b) the traveling salesman follows the goods vehicle, and (c) a third party is engaged for the installation.

(iv) The traveling salesman travels through different vehicles available at nodes (customers' places) for Model 1 (scenario (a) in assumption (iii)) and 2 (scenario (b) in assumption (iii)).

(v) The goods-carrying vehicle is of sufficient capacity.

(vi) Goods are unloaded, and installation is done against some costs and times.

(vii) Halting/penalty charges for the delay in service at nodes are imposed.

(viii) In addition to a general charge depending on distance, goods vehicles charge transporting cost depending on goods' weight and distance.

(ix) In the evolutionary algorithm, (a) three parents concept (assuming surrogate mother) is considered. and (b) the number of children is random (max. number 4 is assumed).

3.3 Traveling salesman problem

In a conventional TSP (2DTSP), a salesperson visits every node exactly once and returns to the starting node, incurring the minimum cost. Consider $\alpha(i, j)$ to be the travel cost from *i*th city to *j*th city. The mathematical model is as follows:

$$\left.\begin{array}{rl} \text{Minimize} & Z = \sum_{i \neq j} \alpha(i, j) x_{ij} \\ \text{subject to} & \sum_{i=1}^{N} x_{ij} = 1 \text{ for } j = 1, 2, \ldots, N \\ & \sum_{j=1}^{N} x_{ij} = 1 \text{ for } i = 1, 2, \ldots, N \end{array}\right\}, \tag{1}$$

with the sub-tour elimination condition

$$\left.\sum_{i \in S}^{N} \sum_{j \in S}^{N} x_{ij} \leq |S| - 1, \forall S \subset P \right\}, \tag{2}$$

where $x_{ij} \in \{0, 1\}, i, j = 1, 2, \ldots, N$, P={1, 2, 3, ..., N} set of nodes, x_{ij} the decision variables, and $x_{ij} = 1$ if the traveling salesman travels from *i*th city to *j*th city; otherwise, $x_{ij} = 0$. The above TSP can be presented as:

$$\left.\begin{array}{rl} \text{determine a complete tour} & (x_1, x_2, \ldots, x_N, x_1) \\ \text{to minimize} & Z = \sum_{i=1}^{N-1} \alpha(x_i, x_{i+1}) + \alpha(x_N, x_1) \\ \text{where} & x_i \neq x_j, i, j = 1, 2, \ldots, N. \end{array}\right\}, \tag{3}$$

along with the sub-tour elimination Eq 2.

3.4 Solid traveling salesman problem or three-dimensional TSP (3D TSP)

In a 3D TSP/solid TSP (STSP), there are several types of vehicles for travel from one node to another. Here, the salesman/service worker chooses the vehicle type, along with the best route to minimize travel costs. Assume $\alpha(i, j, k)$ as the traveling cost from *i*th city to *j*th city using the *k*th type vehicle. The salesperson determines the best tour $(x_1, x_2, \ldots, x_N, x_1)$ with the suitable conveyance $(v_1, v_2, \ldots, v_q)$, where $x_i \in \{1, 2, \ldots, N\}$ for $i = 1, 2, \ldots, N$, $v_k \in \{1, 2, \ldots, q\}$ for $k = 1, 2, \ldots, q$, and all x_i's are distinct. The mathematical formulation of the problem is as follows:

$$\left.\begin{array}{ll} \text{minimize} & Z = \sum_{i=1}^{N-1} \alpha(x_i, x_{i+1}, v_k) + \alpha(x_N, x_1, v_k), \\ \text{where} & x_i \neq x_j, i, j = 1, 2, \ldots, N, \quad v_k \in \{1, 2, \ldots, q\} \end{array}\right\}, \tag{4}$$

along with the sub-tour elimination Eq. 2.

3.5 Solid traveling salesman problem with goods delivery and service (STSPwGDS)

This section addresses the formulation of a time-synchronized TSP problem for goods delivery and their installation by a traveling salesman. It also presents the implication of different vehicles in designing the best route for the traveling salesman. Here, an exclusive car with goods starts from the retailer's godown/warehouse and returns to it, dropping the appropriate article(s) at the customer's locations (nodes) as per the demand placed online. At the same time, the traveling salesman also starts from the warehouse and returns to it after giving services (for installation) to the customers at their locations. He/she uses appropriate conveyance among the available ones at each node. Depending on the arrivals of goods and traveling salesman at a station, the waiting charge for the traveling salesman and demurrage for goods non-clearance are charged. For goods transportation, in addition to normal transportation charges for the vehicle (with goods), an additional amount depending on the goods' weights, is charged. Here, the objective is to find the appropriate travel routes for both goods and traveling salesman so that the total cost of the system (STSPwGDS), including the unloading and service charges, is minimal. To find the best travel routes for both goods vehicle and the traveling salesman (for installation), we develop two different models to understand the differences in the solutions obtained.

Model 1: Goods vehicle and traveling salesman follow best travel routes separately to minimize total costs.

Model 2: Traveling salesman follows the best travel route of the goods vehicle or vice versa.

3.5.1 Mathematical formulation of Models 1 and 2

Model 1:

Following Equation 4, the problem is mathematically formulated as:

$$\left.\begin{aligned} &\text{Minimize} \quad Z=\sum_{i=1}^{N-1}\alpha(x_i,x_{i+1},v_p)\text{dis}(x_i,x_{i+1})\\ &+\alpha(x_N,x_1,v_l)\text{dis}(x_N,x_1)\\ &+\sum_{i=1}^{N}(d_i w_3)\\ &+\sum_{i=1}^{N-1}\beta(y_i,y_{i+1},V)\text{dis}(y_i,y_{i+1})+\beta(y_N,y_1,V)\text{dis}(y_N,y_1)\\ &+\sum_{i=1}^{N-1}(D-\sum_{k=1}^{i}d_k)\text{dis}(y_i,y_{i+1})\xi+\sum_{i=1}^{N}(d_i w_4)\\ &+\sum_{i=1}^{N}\Pi(i)\\ &\text{subject to} \quad \sum_{i=1}^{N}d_i=D\\ &\text{where} \quad p,l\in\{1,2,\ldots,q\}, \quad d_1=0 \end{aligned}\right\}. \tag{5}$$

D represents the total demand (kg) throughout the process, and d_i indicates the demand (kg) for the ith node/location, $d_1 = 0$ because first node represents the depot.

Equation 5 has six parts. In the first part, traveling cost is presented, where $\alpha(x_i, x_{i+1}, v_p)$ indicates the travel cost per unit distance for the traveling salesman from the ith to $(i + 1)$th nodes using the pth vehicle. The second part $(\sum_{i=1}^{N}(d_i w_3))$ represents the total servicing cost incurred by the system, where d_i is the demand of the ith node and w_3 is used for servicing cost per unit weight. In the third part, transportation cost is of goods vehicle only presented, where $\beta(y_i, y_{i+1}, V)$ indicates the transportation cost per unit distance of the goods vehicle, V between the ith and $(i + 1)$th nodes. In the fourth part, $\sum_{i=1}^{N-1}(D-\sum_{k=1}^{i} d_k)\text{dis}(y_i, y_{i+1})\xi$ represents exclusively the transportation cost of goods, which depends on the transported amount between nodes and distance, and ξ indicates the cost per unit distance per unit weight. The cost of transport for goods gradually decreases with the amount of goods. In the fifth part $\sum_{i=1}^{N}(d_i w_4)$, unloading cost is calculated using demand (d_i) and unloading cost per unit weight (w_4). The last part gives the penalty cost for the delay of service and the expression $\Pi(i)$ is represented as:

$$\Pi(i)=\begin{cases}|\theta_i-\phi_i|\eta_1 & :\theta_i>\phi_i\\ |\theta_i-\phi_i|\eta_2 & :\theta_i<\phi_i\\ 0 & :\theta_i=\phi_i\end{cases}$$

Per unit penalty weightages η_1 and η_2 are used for the delayed service corresponding to the traveling salesman and goods vehicle, respectively. Here, θ_i and ϕ_i indicate, respectively, the cumulative times taken by the traveling salesman (to reach) and goods vehicle (to leave) at the ith node.

The cumulative time taken by the transport vehicle to leave the ith node is presented in Eq. 6

$$\left.\begin{aligned} &\phi_1=0 \quad \text{(as journey starts from the depot)}\\ &\phi_i=\sum_{j=1}^{i-1}g(x_j,x_{j+1},V)+\sum_{j=1}^{i}(d_j t_2), \quad i>1\end{aligned}\right\}. \tag{6}$$

Here, $g(x_j, x_{j+1}, V)$ represents the traveling time of the goods vehicle between the jth and $(j + 1)$th nodes and cumulative unloading time upto ith node is represented by $\sum_{j=1}^{i}(d_j t_2)$, where d_j indicates the demand at the jth node and t_2 is the unloading time per unit weight.

The cumulative time taken by the service vehicle to reach the ith node is given in Eq. 7

$$\left.\begin{aligned} &\theta_1=0 \quad \text{(as journey starts from the depot)}\\ &\theta_i=\sum_{j=1}^{i-1}h(x_j,x_{j+1},v_p)\\ &+\sum_{j=1}^{i-1}(d_j t_1)+\Pi^s(i), \quad i>1\\ &\text{where, waiting time of traveling salesman at } i\text{th node}\\ &\Pi^s(i)=|\theta_i-\phi_i|, \theta_i>\phi_i\end{aligned}\right\}. \tag{7}$$

Here, $h(x_j, x_{j+1}, V_p)$ indicates the travel time for the traveling salesman vehicle from jth and $(j+1)$th node using pth vehicle and servicing time up to $(i - 1)$th node represented by $\sum_{j=1}^{i-1}(d_j t_1)$, where t_1 is the service time per unit weight.

Model 2:

In Model 2, both goods and traveling salesman vehicles travel through the same path in visiting the nodes, so the objective function is as follows:

$$\left.\begin{aligned} &\text{Minimize} \quad Z=\sum_{i=1}^{N-1}\alpha(x_i,x_{i+1},v_p)\text{dis}(x_i,x_{i+1})\\ &+\alpha(x_N,x_1,v_l)\text{dis}(x_N,x_1)+\sum_{i=1}^{N}(d_i w_3)\\ &+\sum_{i=1}^{N-1}\beta(x_i,x_{i+1},V)\text{dis}(x_i,x_{i+1})\\ &+\beta(x_N,x_1,V)\text{dis}(x_N,x_1)+\sum_{i=1}^{N-1}(D-\sum_{k=1}^{i}d_k)\\ &\text{dis}(x_i,x_{i+1})\xi+\sum_{i=1}^{N}(d_i w_4)+\sum_{i=1}^{N}\Pi(i)\end{aligned}\right\}, \tag{8}$$

where only one decision variable (x_i) is used as the transporting vehicle follows the traveling salesman vehicle.

3.5.2 Model 1 and 2 with time restrictions (Models 1(a) and 2(a))

This section presents the modified formulation to address the impact of allowable halting times in the proposed

STSPwGDS models. We present a variation by considering constraints on waiting times in some cases, instead of demurrage/stay charges.

In this mathematical formulation, Eq. 5 from Model 1 consists of the following penalty condition.

Let the starting time from depot be t^{st}. Assuming that the whole journey is commenced within a day (24 h),

true cm

current time (t^{ct})= t^{st}+ (Cumulative time (θ_i))

if ($t^{\mathrm{ct}} < t^{\mathrm{tl}}$) and ($\theta_i - \phi_i < \Omega_2$)

$$\Pi(i) = \begin{cases} |\theta_i - \phi_i|\eta_4 + \eta_5 : \theta_i > \phi_i \\ |\theta_i - \phi_i|\eta_2 \quad\quad : \theta_i < \phi_i \\ 0 \quad\quad\quad\quad\quad\quad : \theta_i = \phi_i \end{cases},$$

else

$$= \begin{cases} |\theta_i - \phi_i|\eta_1 : \theta_i > \phi_i \\ |\theta_i - \phi_i|\eta_2 : \theta_i < \phi_i \\ 0 \quad\quad\quad\quad : \theta_i = \phi_i \end{cases},$$

where t^{st}: starting time, t^{ct}: current time, θ_i: cumulative time, t^{tl}: time limit. In this investigation, $t^{\mathrm{st}} = 06:00$, $t^{\mathrm{tl}} = 24$.

Per unit time penalty costs- η_1, η_2 and η_4 are used for the traveling salesman's delay for service. η_1: goods vehicle holding cost per unit time in the night, η_2: traveling salesman stay cost per unit time, η_4: goods vehicle holding cost per unit time in the day as well as waiting time less than maximum day time halt (Ω_2). Additionally, η_5 is used for some fixed charge (unauthorized payment, etc.) for day halt. Ω_2 is the maximum daytime halt of the goods vehicle.

Thus the models-1(a) and -2(a) are, respectively, given by the Eqs. 5 and 8 along with the above time constraints.

3.5.3 STSPwGDS through third-party service (Model 3)

To avoid the delayed arrival of traveling salesman and penalties as consequence, we considered the applicability of the models developed in the presence of a third-party service provider and the fees charged by them. For outsourcing the service, the company assigns it to a third party, instead of its own traveling salesman/service worker. For this work, a charge is paid to the third party. In this consideration, there will be no need for traveling salesman and no demurrage or stay charges. The mathematical formulation of the model is as follows.

$$\left.\begin{array}{l} \text{Minimize} \quad Z = \sum_{i=1}^{N-1} \beta(y_i, y_{i+1}, V)\mathrm{dis}(y_i, y_{i+1}) \\ \quad\quad + \beta(y_N, y_1, V)\mathrm{dis}(y_N, y_1) \\ + \sum_{i=1}^{N-1} (D - \sum_{k=1}^{i} d_k)\mathrm{dis}(y_i, y_{i+1})\xi + \sum_{i=1}^{N} (d_i w_4) \\ \quad + \sum_{i=1}^{N} (d_i w_6) \text{subject to } \sum_{i=1}^{N} d_i = D \end{array}\right\}, \tag{9}$$

where w_6 is the charge per unit demand to be paid to the third party.

4 Varied offspring memetic algorithm (VOMA) to solve the STSPwGDS problem

To develop a new GA, for the solutions of the proposed model we develop a varied offspring concepts into our proposed heuristic. Recently the 'three-parent' concept (cf. Reardon 2017) has been used to combine the DNA from three parents. Following the above phenomena, a variant of memetic algorithm (MA) is developed with the concepts of multi-offspring (including no offspring) and three-parent crossovers, called surro-embryos crossover.

We define our algorithm as a varied offspring memetic algorithm (VOMA) adopting the modified probabilistic selection, a novel varying multi-offspring surro-embryos crossover and a fibo-generation-dependent (Fibo-GD) mutation. The detailed process of the proposed VOMA is represented below.

4.1 Representation

A path i is defined as N-dimensional integer vectors $X_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$, where $x_{i1}, x_{i2}, \ldots, x_{iN}$ indicate N consecutive nodes in a tour. A population with size M is defined by x_{ij}, $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots$, with randomly generated tours using a random number generator function between 1 and N maintaining the TSP conditions. The fitness of a path i is represented by $f(X_i)$ and is evaluated by totaling the costs between the consecutive nodes of the path.

4.2 Selection

4.2.1 Modified Boltzmann probability

We calculate the **Boltzmann probability** (Maity et al. 2015) for each chromosome from the initial population by developing the following expression for the probability of crossover, $p_B = e^{((g/G)*(f_{\min} - f(X_i))/T)}$,
where $T = T_0(1-a)^k$, $k = (1 + C * \mathrm{rand}[0,1])$, $C = \mathrm{rand}[1, 100]$, g =current generation number, G = maximum generation, $T_0 = \mathrm{rand}[60, 150]$, $a = \mathrm{rand}[0, 1]$, and $f(X_i)$ is the objective function.

To form the mating pool, a predefined value, the probability of selection (p_s), is first assigned. For each chromosome of $f(X_i)$, a random number, r, in the range of [0, 1] is generated. If $r < \max(p_s, p_B)$, then the corresponding chromosome is stored in the mating pool, and the chromosome, $f_{\min}$ (minimum fitness of the population), is also taken to the mating pool. Algorithm 1 describes the steps involved. To maintain the exploitation, p_B is defined for each chromosome corresponding to the Boltzmann distribution function, and for the exploration, we use p_s (predefined threshold). Both are used to maintain a good balance between exploration and exploitation.

Algorithm 1: MODIFIED PROBABILISTIC SELECTION PROCEDURE

Input: Total generation number (G), Probability of selection (p_s), pop - size (M), Current generation number (g), $f_{min} = min\{f_1, f_2, \ldots, f_N\}$
Output: Mating pool
for $n \leftarrow 1$ ***to*** M **do**
  r= real random value between 0 and 1
  T_0= integer random value between 60 and 150
  a=real random value between 0 and 1 and C= integer random value between 0 and 100
  k=(1+C*rand[0, 1])
  $T=T_0(1\text{-}a)^k$
  $p_B=e^{((g/G)*(f_{min}-f(X_i))/T)}$
  if $(r < max(p_s, p_B))$ **then**
    choose the corresponding chromosome
  else
    select the chromosome corresponding to f_{min}
  end if
end for

4.3 Crossover

4.3.1 Determination of varying numbers of offspring surro-embryos crossover

In standard GA, two parents generate two offspring, whereas, in nature, the number of offspring born is random, although two parents are involved. VOMA captures this idea to render the offspring and the process more realistic, diversified and competitive. Throughout the process, we maintain the population size (M) constant, even if varying offspring numbers are generated in each generation. If the total offspring numbers are less than M, then a greater number of parents are added. If it is greater than M, then M offspring are retained, based on their fitness values. Also, we apply a three-parent crossover concept. In vitro fertilization (IVF) is a medical treatment procedure in which, in addition to the original parents (father and mother), there is one more mother, known as a surrogate mother, who gestates the offspring(s) in her womb. Inspired by this phenomenon, three-parent crossover is developed to produce offspring in GA for diversity. In the proposed crossover method, we randomly choose three individuals (parents) to produce offspring.

4.3.2 Selection of parents and operationalization of surro-embryos crossover

We chose the number of parents for crossover using the following formula.

Total number of parents (TNOP) = ceil((p_c)*(number of total population)), where p_c is the probability of crossover (given). From the mating pool, we randomly select three parents and create a "parent group" for crossover and continue this formation of groups until eligible parents are available. For producing a child, three number of parents are required, such as father, mother, and surrogated mother so, total number of crossover TNC=floor(TNOP/3). Again, depending on the random offspring number (RON), we produce a different number of offspring with an upper limit of s' offspring.

Next, we elaborate on the three-parent crossover mechanism. Initially, three individuals (parents) are selected randomly from the mating pool, based on a random number between [0, 1]. We select the first three parents (say P_1, P_2, and P_3) following the criterion to $r < p_c$. We illustrate a three-parent crossover example on a five-node TSP in Fig. 1.

Crossover for transportation vehicle:

The three chosen parents are $P_1 = \{23154\}$, $P_2 = \{34512\}$, $P_3 = \{15234\}$. For the first child, we choose randomly a node between 1 and 5 and assume the randomly chosen node is 3. Then, we bring '3' to the beginning of each parent, so the initial parent composition changes to

$P_1^{'} = \{32154\}$, $P_2^{'} = \{34512\}$, $P_3^{'} = \{31524\}$.

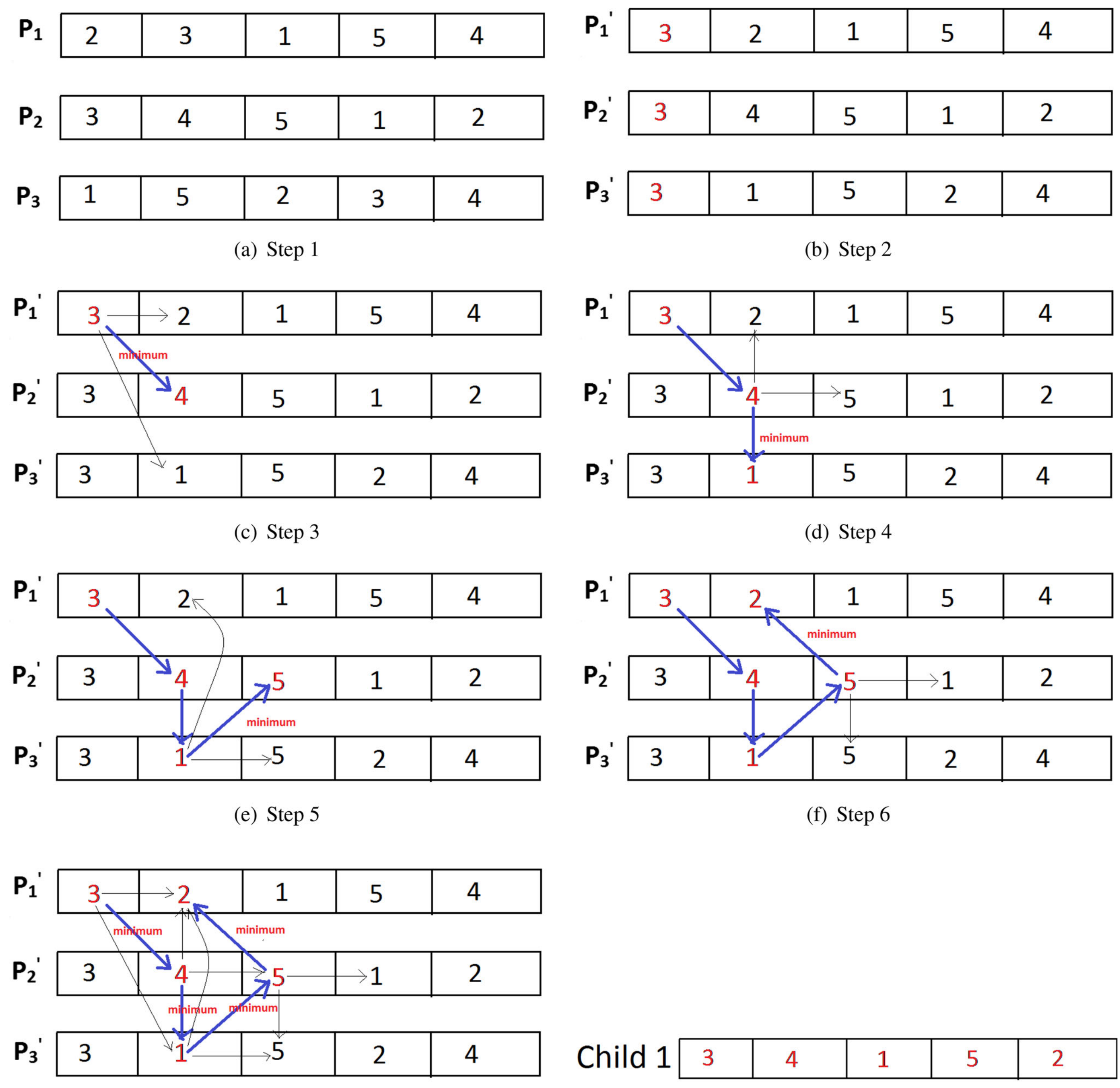
With '3' as the starting node of parents, we identify the next node in the tour by evaluating the costs of arcs (3, 2), (3, 4), and (3, 1) and choose the arc with minimal cost, e.g., (3, 4). We explore the next uncovered nodes from node 4 in P_1, P_2, and P_3, i.e., (4, 2), (4, 5), and (4, 1), to choose the arc with minimal cost, e.g., (4,1). Nodes 3, 4 and 1 are included in the child. We continue the process from node 1 to choose the next uncovered node 5 with minimum cost arc (1,5) after evaluating arcs (1,2) and (1,5). Hence, child's chromosome is created as child 1 = {3 4 1 5 2}.

This process is extended for varied offspring depending upon the random offspring number (RON) by creating child 1 (shown in Fig. 1), child 2 (shown in Fig. 2), child 3, and child 4; i.e., singletons, twins, triplets, and quadruplets are born as different numbers of offspring are produced, and for particular cases depending upon RON (if RON=0), no offspring will be generated.

Based on population size M, a subset of generated offspring is chosen as the next population. The algorithm of the proposed crossover is shown in Sect. 2 with a flowchart in Fig. 4.

Crossover for traveling salesman vehicle:

The traveling salesman/serviceman hires a vehicle at each node out of the available three vehicles at that node to travel to the next node. Hiring charges (travel cost) and velocities (travel times) are different for the different available vehicles.



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4 (e) Step 5 (f) Step 6 (g) All steps at a glance (h) Child 1 produce

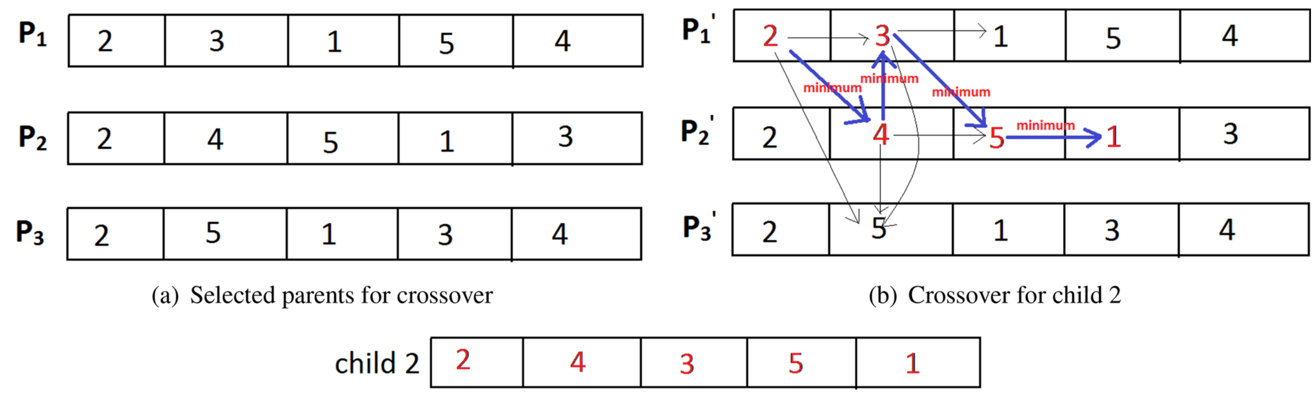
Fig. 1 Surro-embryos crossover for child 1

As traveling salesman uses different types of vehicles, the chromosome representation is different from the goods vehicle. Here, the child is produced using a comparison method focusing on the different vehicle types shown in Fig. 3.

4.4 Mutation

4.4.1 Fibonacci generation-dependent mutation (Fibo-GD)

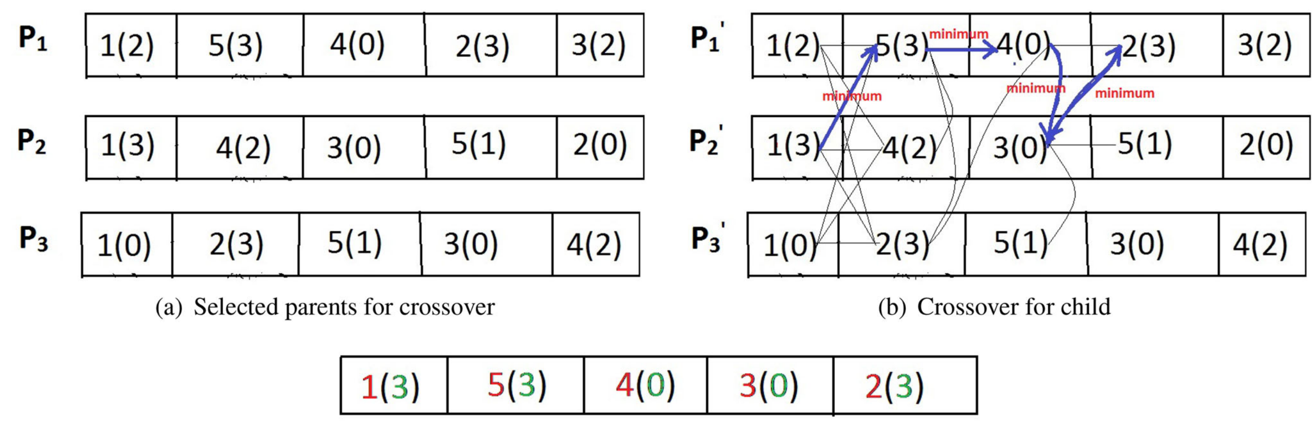
This section elaborates on our modified mutation mechanism. It is expected that values of probability of mutation (p_m) will decrease with the increase in generation. Here, we develop a generation-dependent mutation probability p_m using the concept of the Fibonacci function generating monotonic increasing series, with the mutation probability using the



(a) Selected parents for crossover (b) Crossover for child 2

(c) After crossover produce child 2

Fig. 2 Surro-embryos crossover for child 2



(a) Selected parents for crossover (b) Crossover for child

(c) After crossover produce child

Fig. 3 Surro-embryos crossover for traveling salesman vehicles

inverse of the Fibonacci function being smooth and decreasing monotonically with generation. Thus, p_m is defined as

$$p_m = \frac{k'}{\sqrt{f_g}}, \quad k' \in (0, 1),$$

where f_g is the well-known Fibonacci function, which is represented as $f_g = f_{g-1} + f_{g-2}$ ($f_0 = 0$, $f_1 = 1$ as boundary conditions) with g as the current generation number. The value of k' is set through parameter tuning. For this investigation, it is 0.37. For the first generation, we consider $p_m = 0.2$ and follow the generation-dependent Fibonacci series for subsequent generations.

4.4.2 Mutation process

If $r < p_m$, $r \in$ rand [0, 1], then the corresponding chromosome is selected for mutation. Now, the mutation processes are presented below. Algorithm 3 and Fig. 5 provide a stepwise description of the mutation process. In this mutation, the nodes, alongwith the corresponding vehicles, are interchanged randomly for the traveling salesman. As only one vehicle is used for goods transportation, only nodes are interchanged in the mutation for a goods vehicle. (cf. Fig. 5a, b).

4.5 VOMA algorithm and complexity analysis

A complete stepwise description is provided in Algorithm 4 that summarizes our methodological contribution of a new GA developed for the solutions of the proposed model. We evaluate the time and space complexity of the algorithm below.

4.5.1 Time complexity

A genetic algorithm's complexity, given the standard options (roulette wheel selection, cyclic crossover, random mutation), is O(G(M + MN + MN)), where G is the number of generations, M is the population size, and N is the size of the chromosome/path length. The complexity is there-

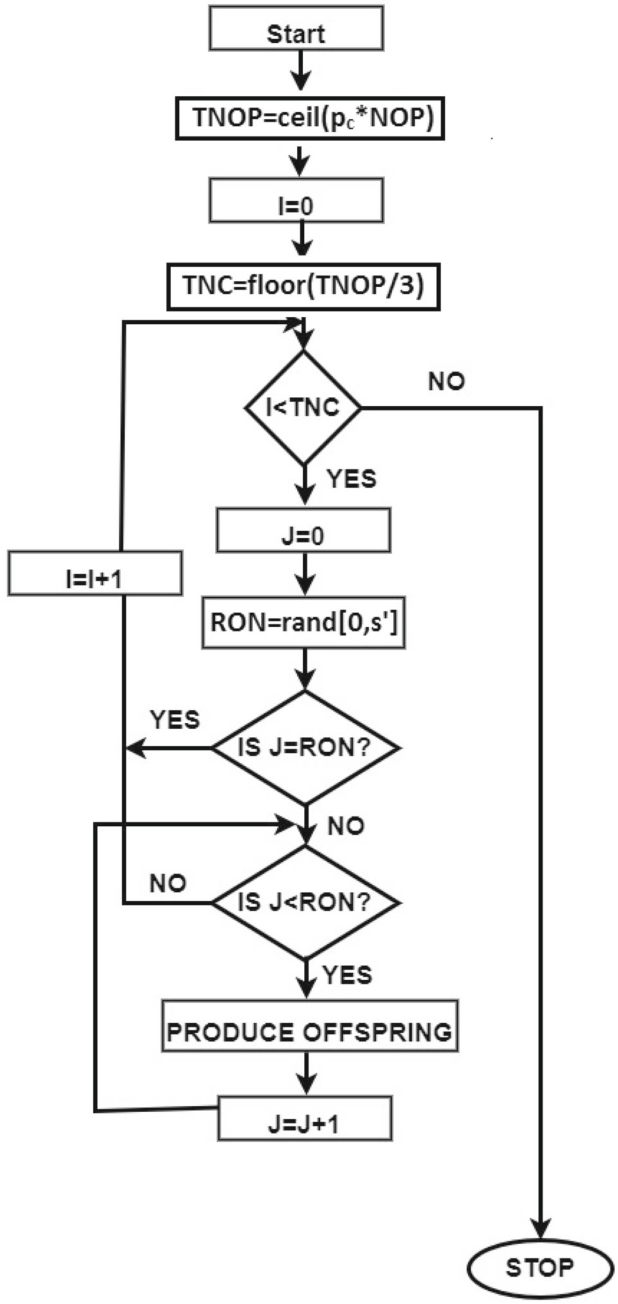


Fig. 4 Flowchart of surro-embryos crossover

Algorithm 2: Surro- Embryos crossover

Input: Selected parents
Output: Offspring

1 total number of parents (TNOP) selected for mating pool = ceil((p_c)*(pop-size))
2 total number of crossovers (TNC) will be = floor(TNOP/3)
3 **for** $i \leftarrow 1$ ***to*** *TNC* **do**
4 choose randomly three distinct parents from mating pool
5 generate random offspring number (RON)=rand[0, s']
6 **for** $j \leftarrow 1$ ***to*** *RON* **do**
7   a_i: generate random number from [0, node]
8   place a_i as the first node of the child's chromosome
9   a_i=min(a_i, each next node)
10   s_1: minimum costs between a_i and each subsequent node of the given parents.
11   replace by s_1 in the next position of the child
12   repeat steps 10 to 12 until the end of the nodes
13 **end for**
14 **end for**

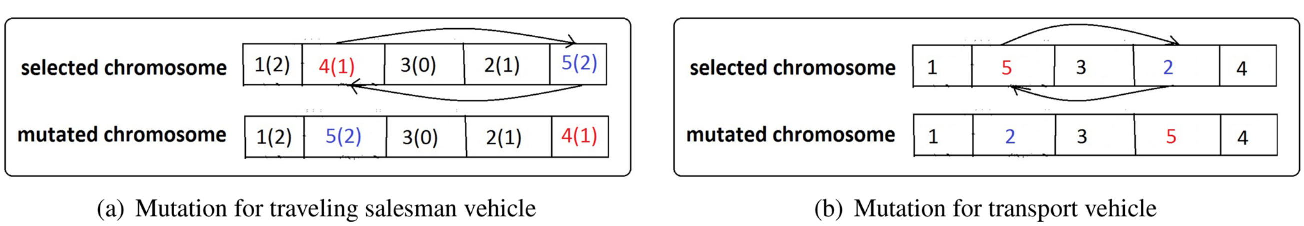
fore on the order of O(GMN). The fitness function, which depends on the application, is being ignored. In the proposed GA (probabilistic selection, surro-embryo crossover, generation dependent mutation) i.e VOMA for routing problem complexity becomes O(G($M + MNF + MN$)), where F is maximum number of offspring produced. Therefore the proposed VOMA has O(GMNF) computational complexity.

4.5.2 Space complexity

In VOMA, the population size (MN) is fixed, so it requires fixed space to save the population. Hence, the space complexity of VOMA is O(MN).

5 Computational experiments

This section primarily focuses on reporting and statistically validating the results obtained using VOMA to establish the solution improvement over standard GA. For experiments, we assumed s'=number of offsprings=4. This value is taken because it is mentioned that 'Families that have 4 or more children come in at just 6 percent' (from the internet). Obviously, a family having more than 4 children will be very



(a) Mutation for traveling salesman vehicle (b) Mutation for transport vehicle

Fig. 5 Mutation for traveling salesman and transport vehicle

Algorithm 3: FIBO- GENERATION- DEPENDENT RANDOM MUTATION

```
Input: Selected chromosome
Output: Mutated chromosome
set g=current generation number
f_0=0, f_1=1, f_g=f_{g-1}+f_{g-2}
if (g = 1) then
    p_m=0.2
else
    p_m=k'/sqrt(f_g), k' ∈(0, 1)
    for i ← 0 to pop − size do
        r=rand(0, 1)
        if (r< p_m) then
            select current chromosome
            a= integer random value between 1 and N
            b= integer random value between 1 and N
            if (a==b) then
                goto 11
            end if
            for j ← 1 to N do
                if (x[j]==a) then
                    p=j
                else if (x[j]==b) then
                    q=j
                end if
                x[p]=b and x[q]=a //replace a with b and b with a
            end for

        end if
    end for
end if
```

Algorithm 4: ALGORITHM FOR PROPOSED VOMA

```
Input: max_ gen, pop-size, p_c, p_s, problem data (distance
       matrix, cost matrix, time matrix, demand matrix, unload
       time and cost matrix, servicing time and cost matrix)
Output: Optimum solutions
g ← 0 // g: iteration/generation number
initialize //according to Sect. 4.1
compute fitness //according to Sect. 4.1
while (g ≤ max_gen) do
    //selection operation
    for i ← 1 to pop − size do
        determine the probability of each chromosome according
        to Sect. 4.2.1
        mating pool produce according to Algorithm 1
    end for
    // crossover operation according to Sect. 4.3.2
    select three parents for crossover using p_c from the mating
    pool
    for i ← 1 to RON do
        modify the parents
        generate offspring according to Algorithm 2
    end for
    // mutation
    generate p_m according to that given in Sect. 4.4.1
    select the offspring for mutations based on p_m
    for i ← 1 to pop − size do
        swap the nodes according to Algorithm 3
    end for
    store the new off springs into offspring set
    build new population
    compute fitness
    sort according to fitness and collect the best chromosomes per
    population size (fixed)
    store the local best solutions
end while
Store the global optimum and near optimum solutions
```

small. For this reason, we have taken a number of children up to four.

5.1 Parametric studies on VOMA

In this section, we summarize the results obtained on the benchmark test data on the bays29 problem set from TSPLIB (cf. Reinelt 1995) to choose the parameter values for running our proposed heuristic. The parameter values are tested using different GA variations, i.e., SGA-I to SGA-VIII, as mentioned in Table 2, to test the robustness. Based on the results obtained, we fix p_c as 0.3 and the maximum number of generations as 2500 for VOMA implementation.

5.2 Performance of VOMA on TSPLIB problems

To judge the effectiveness and feasibility of the developed algorithm VOMA, we applied it to the standard TSP test data sets from TSPLIB (cf. Reinelt 1995). Table 3 reports the results of the said classical TSP starting from 16 nodes to a maximum of 783 nodes by VOMA, SGA-1, MGA (Maity et al. 2015), and ACO (Maity et al. 2017). These results are compared with respect to total cost, iterations, and CPU time in seconds.

VOMA outperformed most of the cases with SGA-1, MGA, and ACO in terms of both solution quality obtained and computational time. The 'Iteration' column in Table 3 indicates the number of iterations that the heuristic requires to reach the best solution. We observe much faster convergence to the best solution for VOMA, and the performance of VOMA continues to be superior for larger problem instances.

5.3 Statistical test

5.3.1 Dispersion against different test problems and different algorithms

Table 4 summarizes the best-known solution (BKS) and average and standard deviation (SD) of results obtained from a particular heuristic over eight instances on a particular benchmark test data set, as well as the error % of the best solution obtained from BKS. The problem sizes of the chosen bench-

Table 2 Parameter analysis

Algorithm	Selection	Crossover	Generation	p_c	p_m	p_s	Result
SGA-I	Roulette wheel	Cyclic	397	0.42	0.31	–	
SGA-II	Tournament	Partial map	432	0.39	0.28	–	
SGA-III	Tournament	Cyclic	256	0.35	0.27	–	
SGA-IV	Rank	Partially map	276	0.32	0.24	–	
SGA-V	Roulette wheel	Surro-Embryos	163	0.30	0.20	0.75	[2020]
SGA-VI	Tournament	Surro-Embryos	182	0.30	0.15	–	
SGA-VII	Rank	Surro-Embryos	173	0.30	0.12	–	
SGA-VIII	Probabilistic	Surro-Embryos	158	0.30	GD		
VOMA	Modified probabilistic	Surro-Embryos	146	0.30	GD	–	
VOMA	Modified probabilistic	Surro-Embryos	132	0.30	Fibo-GD	–	

mark data set vary from 16 nodes to 101 nodes. We also develop four methodological variations of standard GA, i.e., SGA-I, SGA-II, SGA-III, and SGA-IV, to compare their performances with VOMA. In all cases, the average tour distance obtained by VOMA is less than the corresponding average results by SGA-I, SGA-II, SGA-III, and SGA-IV. The SD indicates that these methods are stable and hence emphasizes the robustness of the algorithm. We also obtain the least percentage of relative error in different cases. These errors are also very small, indicating that the average derived solutions are nearer to the BKS. Thus, the proposed VOMA has produced results closer to the best results. Next, we conduct statistical tests to understand the relative difference between the results obtained using VOMA and the other four GA-based algorithms.

5.3.2 Friedman test

To compare the performance of the algorithms SGA-I, SGA-II, SGA-III, SGA-IV, and VOMA, we perform Friedman test (Derrac et al. 2011). It is a non-parametric statistical procedure with the main aim of detecting a significant difference between the behavior of two or more algorithms.

The following assumptions of Friedman test are:

- The results over instances (problems from TSPLIB) are mutually independent (i.e., the results within one instance do not influence the results within other instances).
- Within each instance, the observations (average objective values) can be ranked.

With the assumptions found to be valid, we develop the following hypothesis:

H_0: Each ranking of the algorithms within each problem is equally likely (i.e., there is no difference between them).

H_1: At least one of the algorithms tends to yield larger average objective values than at least one of the other algorithms.

Considering the number of algorithms (k)=5 and the number of instances (b)=8, the Friedman ranking table (Table 5) is prepared based on average values as reported in Table 4.

Consider the expressions of $A_2 = \sum_{i=1}^{b}\sum_{j=1}^{k}[R(X_{ij})]^2$, $R_j = \sum_1^b R(X_{ij})$ for $j = 1, 2, \ldots, k$ and $B_2 = \frac{1}{b}\sum_{j=1}^{k} R_j^2$.

The test statistic is given by: $T_2 = \frac{(b-1)[B_2 - bk(k+1)^2/4]}{A_2 - B_2}$.

From the Table 5, we calculate $A_2 = 443$, $B_2 = 431.75$ and the test statistic $T_2 = 44.64$. The respective F value with a significance level of $\alpha = 0.01$ is $F_{(1-\alpha),(k-1),(b-1)(k-1)} = F_{0.99,4,28} = 4.07$. Since $T_2 > F_{0.99,4,28}$, we reject the null hypothesis. Hence, there exists an algorithm (VOMA), the performance of which is significantly different from the others.

5.3.3 (Post hoc) paired comparisons

If the algorithms a and b are considered different after the rejection of the null hypothesis from Friedman test, following the post hoc paired comparison technique (Derrac et al. 2011), we calculate the absolute differences of the summation of the ranks of algorithms a and b and declare a and b different if:

$$|R_a - R_b| > t_{1-\frac{\alpha}{2}}\left[\frac{2b(A_2 - B_2)}{(b-1)(k-1)}\right]^{\frac{1}{2}},$$

where $t_{1-\frac{\alpha}{2}}$ is the $1 - \frac{\alpha}{2}$ quantile of the t-distribution with $(b-1)(k-1)$ degrees of freedom. Here, $t_{1-\frac{\alpha}{2}}$ for $\alpha = 0.01$ and 28 degrees of freedom is 2.76, and the critical value for the difference is = 17.74.

Table 6 summarizes the paired comparisons, and the underlined values indicate the extent of differences between the algorithms. From Table 6, we conclude that VOMA has outperformed all the other algorithms.

Table 3 Results for standard TSP problems (TSPLIB)

Instances	Best known solution (BKS)	VOMA			SGA-1			MGA (Maity et al. 2015)			ACO (Maity et al. 2017)		
		Cost (Rs.)	Iteration	Time (s)	Cost (Rs.)	Iteration	Time (s)	Cost (Rs.)	Iteration	Time (s)	Cost (Rs.)	Iteration	Time (s)
us16	6859	6859	**47**	0.03	6859	242	0.04	6859	55	0.03	6859	63	0.04
gr17	2085	2085	**64**	0.05	2085	353	0.10	2085	71	0.7	2085	82	0.9
gr21	2707	2707	**148**	0.09	2707	418	0.12	2707	162	0.10	2707	184	0.11
bays29	2020	2020	132	0.14	2020	397	0.42	2020	**61**	0.19	2020	192	0.22
eil51	426	426	**396**	0.51	426	514	1.62	426	399	0.56	426	510	0.59
st70	675	675	341	0.73	810	589	1.83	675	**338**	0.76	691	369	0.88
eil76	538	538	578	0.74	675	635	2.03	538	**543**	0.91	546	612	0.98
eil101	629	703	**764**	1.65	715	926	2.32	632	847	1.97	647	910	2.03
kroA150	26,524	27,105	**873**	1.75	28,396	1098	3.02	27,306	958	2.18	28,146	1012	2.87
kroB150	26,130	26,958	**894**	1.87	29,615	1104	3.18	27,459	1027	2.93	27,659	1095	3.11
a280	2579	3168	**1124**	2.66	3442	1338	4.47	3287	3269	3.28	3198	3287	3.98
lin318	42,029	43,764	**1235**	3.76	44,859	1407	5.85	43,985	1367	4.18	43,857	1397	5.02
pcb442	50,778	52,947	**1407**	4.45	58,107	1693	6.49	53,985	1587	5.28	55,741	1698	6.37
rat783	8806	9726	**1688**	6.19	11,257	2119	7.63	10,246	1846	6.87	11,524	2257	7.85

Table 4 Results of VOMA and other methods

Algorithm	Problem	us16	gr17	gr21	bays29	eil51	st70	eil76	eil101
	BKS⇒	6859	2085	2707	2020	426	675	538	629
	Avg	7132.72	2208.16	2812.56	2143.24	595.62	1061.52	707.15	975.24
SGA-I	SD	8.12	4.31	2.15	7.87	11.41	5.93	10.72	7.31
	Error(%)	3.99	5.90	3.81	6.17	39.81	57.26	31.44	55.04
	Avg	6901.18	2107.82	2731.64	2129.75	514.63	896.28	697.14	912.43
SGA-II	SD	2.89	3.18	1.65	4.92	6.42	4.76	6.12	5.21
	Error(%)	0.61	1.09	0.91	5.49	20.80	32.78	29.57	45.06
	Avg	6876.92	2149.35	2774.52	2089.62	511.58	853.17	693.38	866.71
SGA-III	SD	1.78	3.01	3.12	2.64	2.15	4.01	3.11	4.12
	Error(%)	0.26	3.08	2.49	3.44	20.08	26.39	28.88	37.79
	Avg	6984.32	2187.51	2793.18	2082.38	579.74	1039.36	698.79	949.63
SGA-IV	SD	4.63	2.51	5.61	2.12	4.20	3.82	3.94	6.16
	Error(%)	1.82	4.91	3.18	3.08	36.08	53.97	29.88	50.97
	Avg	6868.95	2093.24	2711.46	2022.16	445.12	729.81	618.63	711.54
VOMA	SD	0.87	0.73	0.49	1.54	1.92	2.61	2.23	3.54
	Error(%)	0.14	0.39	0.16	0.11	4.48	8.10	14.98	13.12

Table 5 Ranking of Friedman test

Algorithms(k) Instances(b)	SGA-I R(X_{b1})	SGA-II R(X_{b2})	SGA-III R(X_{b3})	SGA-IV R(X_{b4})	VOMA R(X_{b5})
us16	5	3	2	4	1
gr17	5	2	3	4	1
gr21	5	2	3	4	1
bays29	5	4	3	2	1
eil51	5	3	2	4	1
eil70	5	3	2	4	1
eil76	5	3	2	4	1
eil101	5	3	2	4	1
Average Rank	5	2.87	2.37	3.75	1
Summation	40	23	19	30	8

Table 6 Paired comparison of Friedman test

$\lvert R_i - R_j \rvert$	SGA-I	SGA-II	SGA-III	SGA-IV	VOMA
SGA-I	–	17	21	10	<u>32</u>
SGA-II	–	–	4	7	<u>15</u>
SGA-III	–	–	–	11	<u>11</u>
SGA-IV	–	–	–	–	<u>22</u>
VOMA	–	–	–	–	–

6 Performance analysis (results for STSPwGDSs)

This section reports the results of VOMA implementation on a 3D TSP with goods delivery and services to understand the relevance of our algorithm by showcasing results from various models.

Our proposed algorithm VOMA has components such as modified probabilistic selection, maiden surro–embryos

Table 7 Model description

Cases	Conditions
Model 1	Individual best paths are determined for the goods vehicle and traveling salesman by minimizing the overall cost
Model 2	Best travel path for goods vehicle is followed by traveling salesman or vice versa
Model 3	Servicing is provided by the third party with the goods vehicle minimizing travel time
Model 1(a)	Here, the goods vehicle is allowed to wait at the roadside for a small charge, instead of waiting at the customer location by paying a penalty to the customer
Model 2(a)	Halt time is considered for Model 2

crossover, and fibo-generation-dependent (Fibo-GD) mutation, and it was implemented in C++ with 150 chromosomes and 2500 generations at maximum. We used a standard Core i5 desktop with 2 GB of RAM to run the code.

6.1 Input data

We furnish the input data in the “Appendix”. Distance matrix, traveling cost per unit distance for the goods vehicle, time matrix for the goods vehicle, traveling costs and times per unit distance for the traveling salesman’s different vehicles, demand matrix, unload time and cost matrices, servicing time and cost matrices, and distance matrix for the goods vehicle for the M/S Sharma Furniture company is presented in Tables 10, 11, 12, 13, 14, 19, 20, 21 and 18, respectively. Data specific to the model parameters are reported in Table 22. Here, we consider three types of vehicles (for traveling salesman (cf. Tables 12, 13)) between two nodes with the corresponding traveling cost and time matrices along different vehicles for the models. For the traveling cost and time (a,b,c) (say), the values a, b, and c are for the 1^{st}, 2^{nd} and 3^{rd} vehicles, respectively.

6.2 Optimum results of STSPwGDS under different models

Table 7 lists the three basic models, along with their variations that we tested on problem instances using VOMA. We present the complete set of results in Tables 8 and 9.

Figure 6 illustrates an overall comparison of all proposed algorithms in terms of total cost and time. Model 1 outperforms Model 2 and hence justifies the requirement of an integrated cost minimization objective while designing the tour instead of following individually best travel tour. Although our results indicate Model 3 as our preferred choice, it is primarily driven by the outsourcing fees offered to the third party. A detailed discussion of the effect of outsourcing fee is provided while discussing Fig. 9.

6.2.1 Discussion of the results

Following Tables 8 and 9, total unloading and servicing costs remain unaltered in Models 1 and 2, with Model 1 emerging as the preferred choice considering the total cost. Further elaboration of total cost reveals that Model 1 incurs a transportation cost of the goods vehicle of \$1047.91 (path: 0–5–1–9–4–8–3–2–6–7) and a traveling salesman cost of \$307.58 (path and vehicle types in parentheses: 0(0)–5(1)–9(2)–1(2)–4(1)–8(1)–3(2)–2(2)–6(2)–7(2)). For Model 2, the corresponding goods vehicle cost is \$1082.58 (path: 0–2–6–7–1–9–4–8–5–3), and the traveling salesman cost is \$253.67 (path: 0(2)–2(1)–6(2)–7(0)–1(0)–9(0)–4(0)–8(0)–5(2)–3(1)). Although Model 2 is able to produce a lower transportation cost of \$1329 vis-s-vis Model 1 with a cost of \$1355, it does not yield the lowest overall cost, i.e., the cost of \$2436 for Model 2 against the cost of \$2311 for Model 1.

If we allow the goods vehicle to halt at some nodes during the daytime for up to 5 h in both Models 1 and 2, then from Table 9, Model 1(a) yields a lesser total cost (\$2310.34) than Model 2(a) (\$2428.81).

Figure 6 shows all of the models’ costs and times.

Moreover, the total cost in Model 1(a) (\$2310.34) is marginally less than in Model 1 (\$2311.39) for the present set of data. It should be pointed out that the total operational times for Models 1 and 1(a) are the same, although the total operation time for the goods vehicle is greater in Model 1(a), due to the brief halts by the goods vehicle at some nodes. In Model 1, during these halt periods, the goods were in storehouses, which was not considered in calculating the total operation time of the goods vehicle.

Similar is the case for Models 2 and 2(a). It is interesting to note that, in Table 9 for Model 1(a), at node 7, the halting of the goods vehicle is not allowed since the daytime is over by only 0.04 h, increasing the demurrage cost for the system. Since we have considered a crisp data set, such marginal cases are not appropriately adjusted.

Since the third-party installation removes many costs, such as the traveling salesman’s travel and stays costs, demurrage costs for goods, this model, i.e., Model 3 furnishes the route as (0–5–3–2–6–7–1–9–4–8) and the total cost as \$861.81 plus the service cost by the third party. Obviously, the above route is the goods vehicle’s minimum cost route, and the total cost depends on the service charge per unit. Assuming a service charge of \$60 per unit weight for third-party servicing, i.e., $w_6 = \$60$ and total servicing cost by third-party = 9.4* \$60 = \$564, the total cost comes to (\$861.81 + \$564) = \$1425.81. Additionally, we assume that third-party servicing time is the same as before, i.e., 18.8 h (Model 3). Hence, the total time is taken = (403.4+18.8) h = 422.2 h, which is still less than with Model 1(a) because the total service charge is still less than the traveling salesman’s travel and stay costs and demurrage costs for goods against his or her own servicing. Therefore, from this criteria, management can calculate the maximum possible allowed third-party service charge.

6.2.2 Parametric analysis for STSPwGDS

For Model 1 with different numbers of chromosomes (noc) and iterations, the best results are obtained and are presented graphically in Fig. 7. In both cases, the best value reduces with decreasing returns. Figure 7a, b illustrate the effects on total cost with generation number and number of chromosomes, respectively.

Table 8 Best results of Model 1

Model	Goods vehicle TVS vehicle	Path Path	0 0(0)	5 5(1)	1 9(2)	9 1(2)	4 4(1)	8 8(1)	3 3(2)	2 2(2)	6 6(2)	7 7(2)	Total
		Time	24	20	47	39	41	26	45	43	40	30	355
	Transportation	Cost	48	240	32	88.05	72	199.92	56	72	80.04	159.9	1047.91
		Time	–	–	–	–	–	–	–	–	–	–	–
	Goods	Cost	–	9.40	8.00	6.50	5.50	4.40	3.60	2.90	2.00	1.40	43.70
		Time	–	1.4	1.5	1.0	1.1	0.8	0.7	0.9	0.6	1.4	9.4
1	Unloading	Cost	-	2.80	3.00	2.00	2.20	1.60	1.40	1.80	1.20	2.80	18.80
		Time	10	17	17	35	22	12	45	38	41	9	246
	Traveling	Cost	13	54.4	54.4	36.96	19.2	39.2	11.12	12.10	13.20	54	307.58
		Time	–	0/15.4	67/49.7	0/0	16.9/0	0/.7	0/13.1	.5/0	0/3.8	.8/0	85.2/82.7
	Holding/stay	Cost	-	0/92.40	268/298.20	0/0	67.60/0	0/4.20	0/78.60	2/0	0/22.80	3.20/0	340.80/496.20
		Time	–	2.8	3	2	2.2	1.6	1.4	1.8	1.2	2.8	18.8
	Servicing	cost	–	8.40	9.00	6.00	6.60	4.80	4.20	5.40	3.60	8.40	56.40
		Time											797.1
	Total	Cost											2311.39
		TPD	10	32	5	15	12	28	8	10	12	30	162
	Total	TVD	10	32	32	33	12	28	8	10	12	30	207

TPD transportation distance, *TVD* traveling distance, *TVS* traveling salesman

Table 9 Best results of Models 2, 3, 1(a), and 2(a)

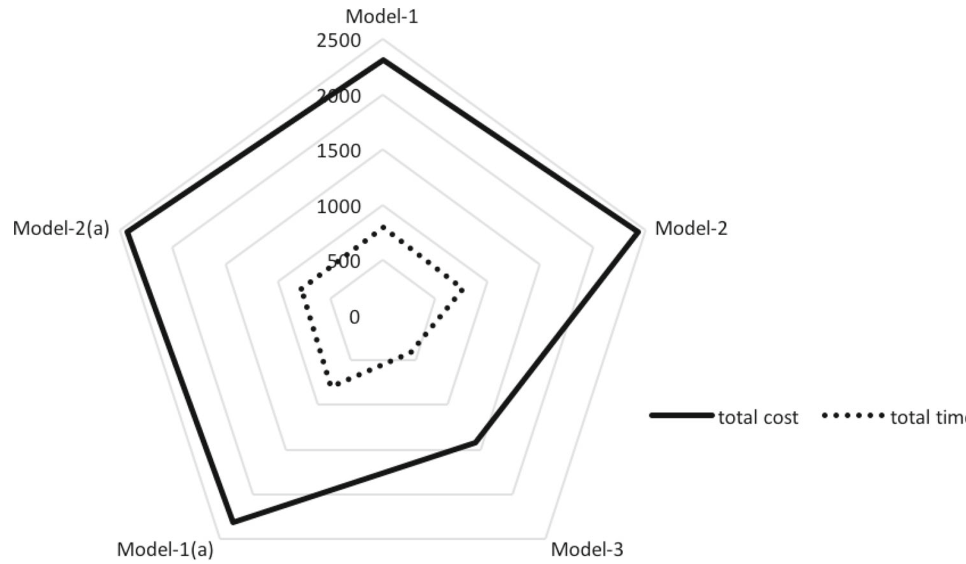
Model	Goods/TVS	Path	0(2)	2(1)	6(2)	7(0)	1(0)	9(0)	4(0)	8(0)	5(2)	3(1)	Total
		Time	22	43	40	49	47	39	41	28	46	23	378
	Transportation	Cost	224	82.4	80.04	40.02	32	88.05	72	175.95	64	224.12	1082.58
		Time	–	–	–	–	–	–	–	–	–	–	–
	Goods	Cost	–	9.40	8.50	7.90	6.50	5.00	4	2.90	2.10	0.70	47.00
		Time	–	0.9	0.6	1.4	1.5	1.0	1.1	.8	1.4	.7	9.4
2	Unloading	Cost	–	1.8	1.2	2.8	3.0	2.0	2.2	1.6	2.8	1.4	18.80
		Time	14	36	19	15	21	20	18	13	28	15	199
	Traveling	Cost	61.25	13.1	13.2	8.4	5.5	24	22.8	43.7	17	44.72	253.67
		Time	–	0/8.9	0/5.8	0/21.2	0/32.7	0/24	0/18.1	0/21.6	0/14.8	0/15.9	0/163
	Holding/stay	Cost	–	0/53.4	0/34.8	0/127.2	0/196.2	0/144	0/108.6	0/129.6	0/88.8	0/95.4	0/978
		Time	–	1.8	1.2	2.8	3.0	2.0	2.2	1.6	2.8	1.4	18.8
	Servicing	Cost	–	5.40	3.60	8.40	9.00	6.00	6.60	4.80	8.40	4.20	56.40
		Time											768.2
	Total	Cost											2436.45
	Total	TPD/TVD	35	10	12	6	5	15	12	23	10	26	154
	Goods	Path	0	5	3	2	6	7	1	9	4	8	total
		Time	24	46	45	43	40	49	47	39	41	20	394
	Transportation	Cost	48	64	60	72	80.04	40.02	32	88.05	72	240	796.11
		Time	–	–	–	–	–	–	–	–	–	–	–
	Goods	Cost	–	9.40	8.00	7.30	6.40	5.80	4.40	2.90	1.90	0.8	46.90
		Time	–	1.4	0.7	.9	.6	1.4	1.5	1.0	1.1	0.8	9.4
3	Unloading	Cost	–	2.80	1.40	1.80	1.20	2.80	3.00	2.00	2.20	1.60	18.8
		Time	–	–	–	–	–	–	–	–	–	–	–
	Traveling	Cost	–	–	–	–	–	–	–	–	–	–	–
		Time	–	–	–	–	–	–	–	–	–	–	–
	Holding/stay	Cost	–	–	–	–	–	–	–	–	–	–	–
	by third-party	Time	–	2.8	1.4	1.8	1.2	2.8	3.0	2.0	2.2	1.6	18.8
	Servicing	Cost	–	84	42	54	36	84	90	60	66	48	564
		Time											422.2
	Total	Cost											1425.81
	Total	TPD/TVD	10	10	8	10	12	6	5	15	12	32	120
	Goods vehicle	Path	0	5	1	9	4	8	3	2	6	7	total

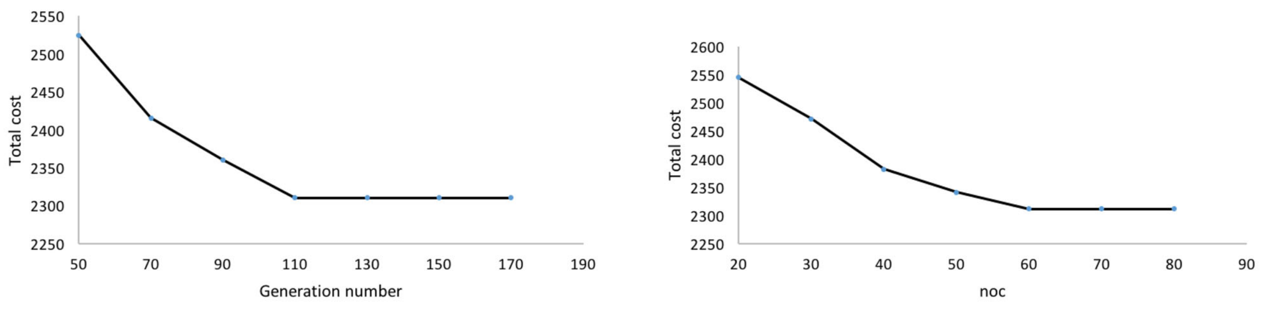
Table 9 continued

Model	Goods/TVS	Path	0(2)	2(1)	6(2)	7(0)	1(0)	9(0)	4(0)	8(0)	5(2)	3(1)	Total
	TVS vehicle	Path	0(0)	5(1)	9(2)	1(2)	4(1)	8(1)	3(2)	2(2)	6(2)	7(2)	Total
		Time	24	20	47	39	41	26	45	43	40	30	355
	Transportation	Cost	48	240	32	88.05	72	199.92	56	72	80.04	159.9	1047.91
		Time	–	–	–	–	–	–	–	–	–	–	–
	Goods	Cost	–	9.40	8.00	6.50	5.50	4.40	3.60	2.90	2.00	1.40	43.70
		Time	–	1.4	1.5	1.0	1.1	0.8	0.7	0.9	0.6	1.4	9.4
1(a)	Unloading	Cost	–	2.80	3.00	2.00	2.20	1.60	1.40	1.80	1.20	2.80	18.80
		Time	10	17	17	35	22	12	45	38	41	9	246
	Traveling	Cost	13	54.4	54.4	36.96	19.2	39.2	11.12	12.10	13.20	54	307.58
		Time	–	0/15.4	67/49.7	0/0	16.9/0	0/.7	0/13.1	.5/0	0/3.8	.8/0	85.2/82.7
	Holding/stay	Cost	–	0/	268/	0/	67.60/	0/	0/	(0.5+0.45)/	0/	3.20/	0.95+338.8/
				92.40	298.20	0	0	4.2	78.60	0	22.80	0	496.20
		Time	–	2.8	3	2	2.2	1.6	1.4	1.8	1.2	2.8	18.8
	Servicing	Cost	–	8.40	9.00	6.00	6.60	4.80	4.20	5.40	3.60	8.40	56.40
		Time											797.1
	Total	Cost											2310.34
		TPD	10	32	5	15	12	28	8	10	12	30	162
	Total	TVD	10	32	32	33	12	28	8	10	12	30	207
	Goods/service worker	Path	0(2)	2(1)	6(2)	7(0)	1(0)	9(0)	4(0)	8(0)	5(2)	3(2)	total
		Time	22	43	40	49	47	39	41	28	46	23	378
	Transportation	Cost	224	82.4	80.04	40.02	32	88.05	72	175.95	64	224.12	1082.58
		Time	–	–	–	–	–	–	–	–	–	–	–
	Goods	Cost	–	9.40	8.50	7.90	6.50	5.00	4	2.90	2.10	0.70	47.00
		Time	–	0.9	0.6	1.4	1.5	1.0	1.1	.8	1.4	.7	9.4
2(a)	Unloading	Cost	–	1.8	1.2	2.8	3.0	2.0	2.2	1.6	2.8	1.4	18.80
		Time	14	36	19	15	21	20	18	13	28	18	202
	Traveling	Cost	61.25	13.1	13.2	8.4	5.5	24	22.8	43.7	17	37.18	246.13
		Time	–	0/8.9	0/5.8	0/21.2	0/32.7	0/24	0/18.1	0/21.6	0/14.8	0/15.9	0/163
	Holding/stay	Cost	–	0/53.4	0/34.8	0/127.2	0/196.2	0/144	0/108.6	0/129.6	0/88.8	0/95.4	0/978
		Time	–	1.8	1.2	2.8	3.0	2.0	2.2	1.6	2.8	1.4	18.8
	Servicing	Cost	–	5.40	3.60	8.40	9.00	6.00	6.60	4.80	8.40	4.20	56.40
		Time											771.2
	Total	Cost											2428.81
	Total	TPD/TVD	35	10	12	6	5	15	12	23	10	26	154

TPD transportation distance, *TVD* traveling distance, *TVS* traveling salesman

Fig. 6 Total costs and travel times of all models





(a) Best values vs. generation number in Model 1 (no. of chromosomes=30)

(b) Best values vs. no. of chromosomes in Model 1 (generation number=100)

Fig. 7 Parameter analysis of Model 1

7 Practical implementation

This section illustrates one practical implementation of the algorithms developed. We chose a furniture dealer named M/S Sharma Furniture, located in Kharagpur, West Bengal, India. The company collects orders throughout the year and supplies the materials to distant customers quarterly by lot. We collect customer data on location and demand. Transportation cost is evaluated from a distance measured through Google Maps, and we approximate loading/unloading costs, service time, and penalty/demurrage information from information captured from the company database. Interested readers can refer to the "Appendix" for further details.

We run both Models 1 and 2 on this problem context with ten customer locations to understand the implications. Figure 8 shows separate routes for the goods vehicle and salesperson following Model 1 and the best route for the transport vehicle followed by the salesperson in Model 2.

We solve Model 1 using VOMA to obtain the total best distance for the salesperson vehicle (red line) as 317.88 km, with the route as Kharagpur(0)–Debra(2)–Midnapur(1)–Keshpur(9)–Salbani(4)–Binpur(8)–Jhargram(5) –Gopiballabpur(7)–Datan(6)–Sabang(3)–Kharagpur(0). Model 1 yields the total distance for the goods vehicle (purple line) as 309.04 km, with the route as Kharagpur(0)–Sabang(3)–Debra(2)–Midnapur(1)–Keshpur(9)–Salbani(4) –Binpur(8)–Jhargram(5)–Gopiballabpur(7)–Datan(6)–Kharagpur(0). Model 2 obtains the distance of the corresponding route (black line) of 292.32 km, and the route is Kharagpur(0)–Midnapur(1)–Keshpur(9)–Salbani(4)–Binpur(8)–Jhargram(5) –Gopiballabpur(7)–Datan(6)–Sabang(3)–Debra(2)–Kharagpur(0).

We observe that, for Model 1, the salesperson vehicle and goods vehicle travel 317.88 km and 309.04 km, respectively. In contrast, for Model 2, both the salesperson vehicle and goods vehicle travel 292.32 km, which is less than in Model 1. However, if we consider the traveling cost of the salesperson, transportation cost of the goods vehicle, unloading cost, penalty cost, and servicing cost, then the overall best cost

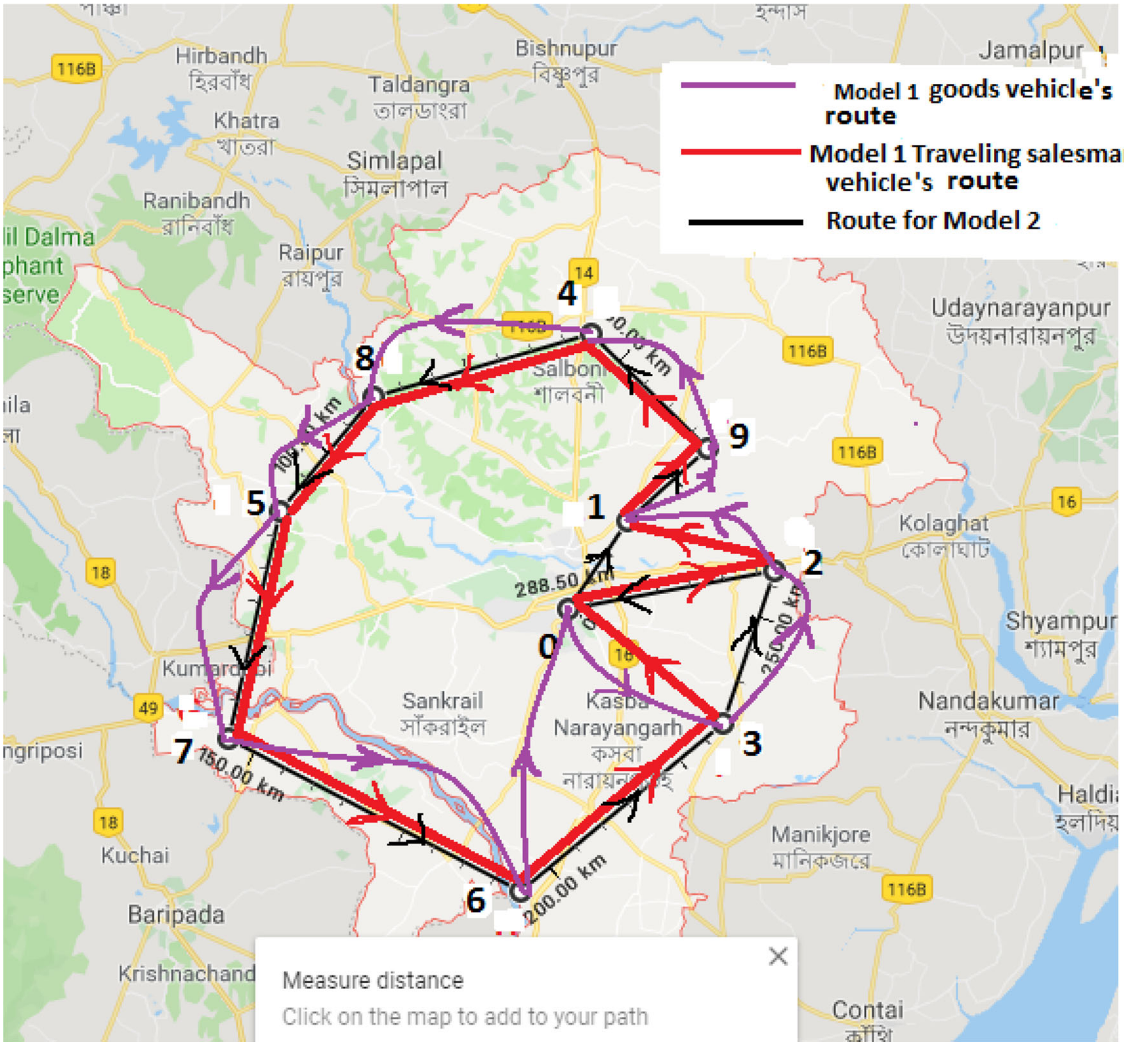


Fig. 8 Best paths with respect to Models 1 and 2 for the Shrama Furniture Company

for Model 1 (\$3493.56) is less than the overall best cost for Model 2 (\$3587.48).

outsourcing the service part to a third party depending on the service charge or bargaining cost per unit. We understand that the maximum amount to be paid for third-party service can be calculated using equation 10 as follows.

$$w_6 \leq \left. \frac{\sum_{i=1}^{N-1} \alpha(x_i, x_{i+1}, v_p) * \mathrm{dis}(x_i, x_{i+1}) + \alpha(x_N, x_1, v_l) * \mathrm{dis}(x_N, x_1) + \sum_{i=1}^{N} \Pi(i) + \sum_{i=1}^{N} (d_i * w_3)}{D} \right\} . \tag{10}$$

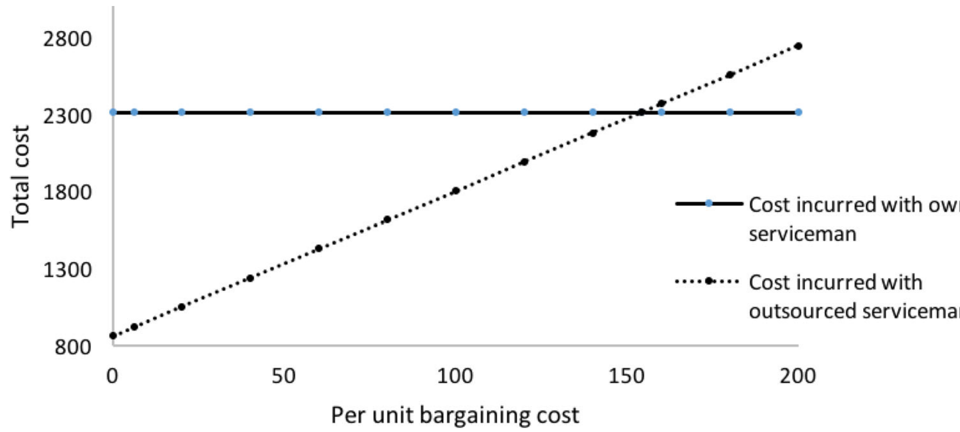
(Max. third-party service charge per unit weight ≤ (traveling salesman's travel cost+demurrage and stay costs+service cost)/(average demand across nodes)).

Figure 9 illustrates the change in decision-making with changes in per unit negotiated service charge with the third-party agency. With an increase in the outsourcing service charge, the total cost continues to increase, and it becomes more profitable to opt for Model 1(a) by discontinuing the outsourcing model, i.e., Model 3.

8 Managerial insight

This section addresses the impact of the parameters on total cost and the associated shift in the decision-making process. Models 1 and 2, along with their variations, i.e., Models 1(a) and 2(a), cover possible scenarios and solutions to the decision problems that can arise in situations specific to STSPwSGDS. Model 3 addresses the decision problem of

Fig. 9 Change in total cost with third-party service charge



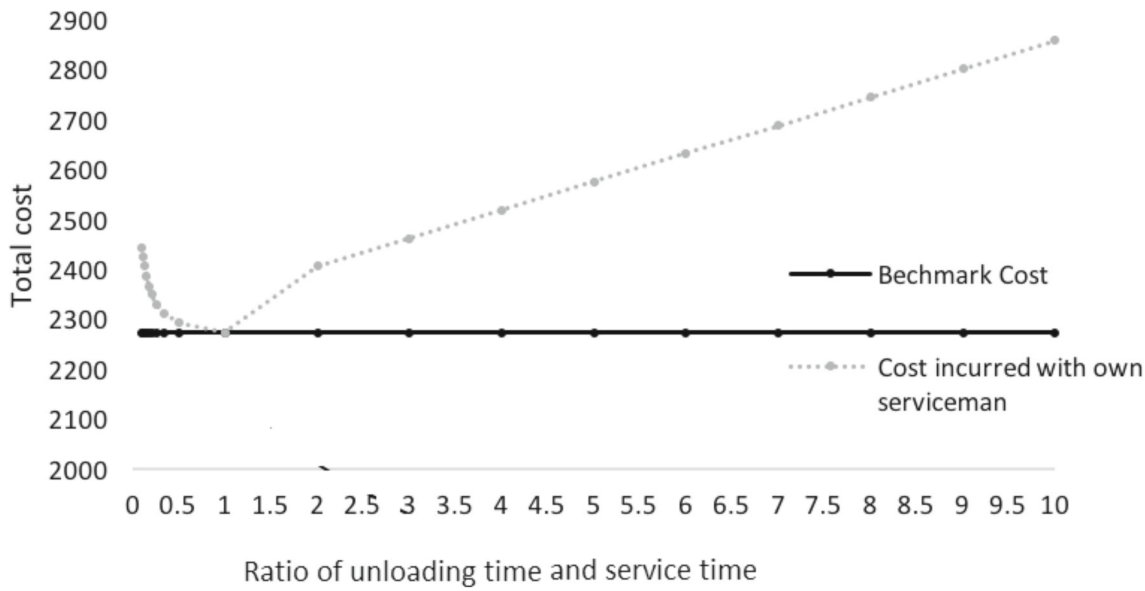
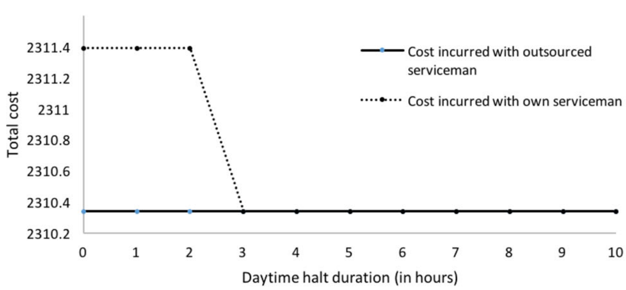
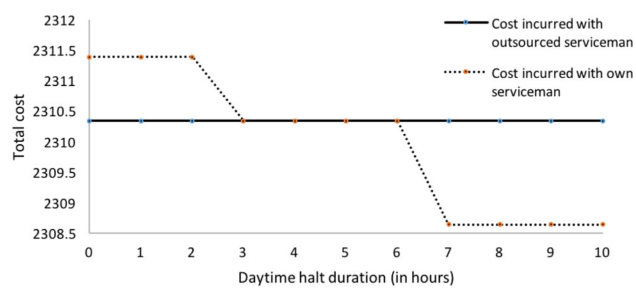


Fig. 10 Ratio of unloading time and servicing time vs. total cost



(a) daytime halt (in hrs.) vs. total cost



(b) daytime halt (in hrs.) vs. total cost

Fig. 11 Change in best cost with allowable halt duration and time

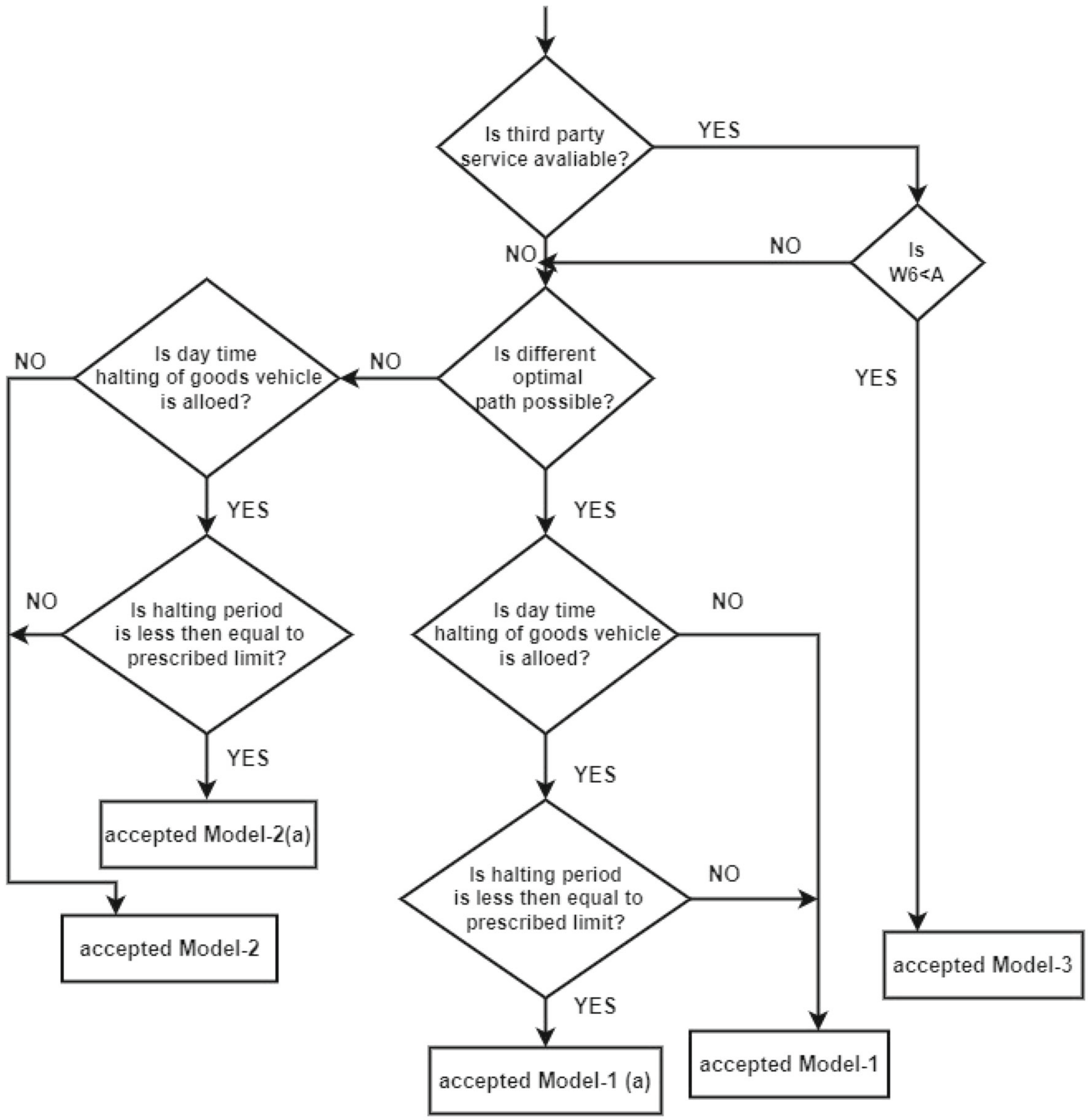


Fig. 12 Management decision flowchart

Next, we understand the influence of other problem context parameters for deciding whether to choose outsourcing (Model 3) or to choose Model 1(a). We chose two important parameters that create an effective trade-off between outsourcing and following Model 1. The ratio of unloading time and service time indicates the amount of waiting that either the goods vehicle or the traveling salesman must do to complete the product delivery. When the ratio is 1, it indicates an ideal balance between these two. If the ratio is less than 1, the service time becomes longer than the unloading time; i.e., the goods vehicle must wait. Similar to higher ratio values, the traveling salesman must wait for the goods vehicle to arrive. Figure 10 indicates the change in total cost with the ratio value. The benchmark cost line in Fig. 10 indicates the cost in the best-case scenario, i.e., when the ratio value is 1.

With an opportunity to halt, Model 1(a) provides a superior result and will be preferred over Model 1. Increasing the halting time allows the goods vehicle to synchronize the arrival time with the arrival of the traveling salesman at customer locations. Figure 11 shows the reduction in total cost with increasing halt duration and the resulting switch in decision-making from Model 3 to Model 1(a). In Fig. 11a, the end time is restricted until 6 PM, whereas in Fig. 11b, the end time is extended until 7 PM. Please note that the halt time is not extended beyond the end time, and the goods vehicle must wait at the customer's location by paying the penalty cost. The total cost in an outsourcing model is considered a benchmark. The extension of the halting time from 6 PM to 7 PM is observed with a reduction in total cost.

Next, we focus on the decision-making situations between Models 1 and 2. If halting is not allowed, we opt for Model 1, in which the overall cost is minimized by allowing the goods vehicle and traveling salesman to take different routes. Model 2(a), followed by Model 2 will be chosen if different routes are not allowed by goods vehicle and traveling salesman for some firm-specific constraint.

Figure 12 summarizes this discussion by providing a flowchart capturing all of the scenarios or a real-life STSPwGDS problem to obtain a better solution in terms of cost. This outcome provides a clearly defined managerial decision-making framework that managers can use to make rational decisions with the objective of minimizing total costs. These decisions are derived from one experiment only. To make these as the general decisions, some more experiments are required.

9 Conclusion and future scope

This paper develops a new memetic algorithm, VOMA, with modified probabilistic selection. A new, varied offspring three-parent crossover and fibo-generation-dependent mutation are developed and implemented successfully for STSPwGDS problems. Here, we develop a generic formulation that can be used in various associated problem contexts. In STSPwGDS, the delivery vehicle and the traveling salesman for installation can move separately or together (one can follow the other). Foe the usefulness of developing two distinct tours by minimizing total costs, we develop two models, Models 1 and 2, to compare the results, and we conclude that Model 1 yields better costs than Model 2, although Model 2 has a shorter distance traveled by the goods vehicle and traveling salesman. The solution obtained by our algorithm also illustrates the utility of multiple vehicle types, with which the traveling salesman chooses a vehicle type even with a higher per unit cost to reduce the penalty cost paid to the customer. To justify the consideration of different vehicles for the traveling salesman journey, the results obtained by our algorithm showcase an effective trade-off between various cost components and vehicle types. The decision-making framework, along with the development of two model variations by incorporating halting time restrictions, effectively addresses the obvious possibility of the late arrival of the traveling salesman and its effect (penalty) on the total cost. The possibility of having the servicing activity outsourced is also considered. Our paper concludes that the propensity to outsource will decrease with increasing service charges and halt times, but the decision-maker will tend to choose outsourcing if the values of unloading time and service time increase. It also provides a ready-to-use decision-making framework for practitioners to use. We substantiate this claim by effectively implementing our algorithm on a practical problem instance.

One limitation of the present investigation is the crisp nature of the formulation and the numerical illustration of STSPwGDS models. If the input data are of the interval or fuzzy type, then the aforementioned case could be more pragmatic and should be addressed appropriately. Therefore, the proposed STSPwGDS can be formulated and solved with imprecise parameters and data, i.e., fuzzy, rough, fuzzy–random, etc.

Funding The authors have not disclosed any funding.

Data availability Data available on request.

Declarations

Conflict of interest The authors certify that there is no conflict of interest with any individual/organization for the present work.

Human participants This paper does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in this study.

Table 10 Input data: distance matrix

i/j	0	1	2	3	4	5	6	7	8	9
0	∞	26	35	26	25	10	38	40	45	35
1	32	∞	22	30	33	44	32	40	24	5
2	40	32	∞	32	24	34	10	34	42	32
3	26	32	8	∞	22	28	28	32	26	38
4	39	24	37	32	∞	25	26	32	12	26
5	27	32	28	10	26	∞	30	42	30	32
6	29	27	32	24	39	42	∞	12	30	22
7	30	6	34	44	38	27	42	∞	20	36
8	32	38	37	28	21	23	35	30	∞	26
9	31	32	30	25	15	30	36	42	32	∞

Appendix A Input data

Here, we have taken the distance matrix, transportation cost and time per unit distance, traveling cost and time per unit distance, predefined demand/requirement, unload time and cost, servicing time and cost at every node and three types of traveling salesman vehicle and only one type of goods vehicle are considered. Also values of the parameters for STSPwGDS are presented in Tables 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 and 22.

Table 11 Input data: traveling cost per unit distance of goods vehicle

i/j	0	1	2	3	4	5	6	7	8	9
0	∞	7.69	6.4	9.85	6.4	4.8	7.37	7.4	7.11	6.8
1	9.25	∞	7.27	7.47	8.48	7.27	7.5	8.4	9.33	6.4
2	8.4	7	∞	7.5	8.33	8.24	7.2	7.53	7.62	7.5
3	8.62	7.5	7	∞	7.27	7.14	8.57	8.75	6.77	7.79
4	7.59	7.33	7.57	7.5	∞	6.4	7.69	7.5	6	8.62
5	7.41	7.5	7.14	6.4	8.62	∞	8.53	7.62	8.53	7.5
6	7.72	7.41	7.5	7.33	7.59	7.62	∞	6.67	8.53	7.27
7	5.33	6.67	7.53	7.27	7.37	7.41	7.62	∞	8.8	8.22
8	7.5	8.42	7.57	7.14	7.62	7.65	8.46	8.53	∞	8.62
9	7.23	7.5	7.47	6.4	5.87	8.53	8.22	7.62	7.5	∞

Table 12 Input data: time matrix for goods vehicle

i/j	0	1	2	3	4	5	6	7	8	9
0	∞	27	22	20	31	24	15	14	12	20
1	14	∞	34	22	17	10	20	9	23	47
2	9	24	∞	22	25	19	43	22	13	22
3	23	24	45	∞	33	26	20	18	30	14
4	17	31	15	23	∞	30	28	22	41	22
5	28	20	28	46	26	∞	22	14	18	20
6	22	28	22	28	16	11	∞	40	22	30
7	30	49	19	10	18	25	10	∞	30	16
8	20	10	19	26	32	28	15	22	∞	25
9	24	22	22	32	39	18	14	11	21	∞

Table 13 Input data: traveling cost per unit distance of traveling salesman vehicle

i/j	Crisp travel cost per unit distance matrix(10×10) with three conveyances									
	0	1	2	3	4	5	6	7	8	9
0	∞	(.75,1.19,1.16)	(1.71,1.2,1.75)	(1.73,1.85,1.62)	(1.12,1.00,1.2)	(1.3,0.8,0.9)	(1.37,1.27,1.07)	(1.33,1.3,1.53)	(1.27,1.22,1.29)	(1.26,1.17,1.20)
1	(1.42,1.67,1.15)	∞	(1.11,1.78,1.89)	(1.61,1.42,1.77)	(1.31,1.62,1.12)	(1.6,1.42,1.52)	(1.81,1.62,1.91)	(1.32,1.41,1.6)	(1.6,1.2,1.81)	(1.1,1.4,1.6)
2	(1.42,1.63,1.23)	(1.78,1.45,1.62)	∞	(1.67,1.21,1.42)	(1.81,1.73,1.54)	(1.67,1.52,1.35)	(1.43,1.31,1.21)	(1.87,1.69,1.45)	(1.53,1.32,1.12)	(1.81,1.67,1.42)
3	(1.61,1.72,1.43)	(1.67,1.12,1.41)	(1.56,1.62,1.39)	∞	(1.68,1.82,1.79)	(1.18,1.31,1.49)	(1.61,1.42,1.72)	(1.81,1.71,1.52)	(1.67,1.31,1.42)	(1.62,1.71,1.42)
4	(1.53,1.72,1.32)	(1.61,1.48,1.82)	(1.68,1.51,1.71)	(1.81,1.68,1.42)	∞	(1.4,1.67,1.12)	(1.7,1.5,1.2)	(1.8,1.4,1.3)	(1.9,1.6,1.4)	(1.8,1.6,1.42)
5	(1.7,1.6,1.4)	(1.7,1.5,1.2)	(1.6,1.8,1.4)	(1.4,1.6,1.7)	(1.8,1.7,1.5)	∞	(1.8,1.4,1.7)	(1.6,1.3,1.5)	(1.7,1.5,1.8)	(1.5,1.7,1.9)
6	(1.8,1.6,1.7)	(1.8,1.6,1.7)	(1.8,1.6,1.4)	(1.81,1.6,1.3)	(1.72,1.6,1.4)	(1.6,1.7,1.4)	∞	(1.2,1.4,1.1)	(1.6,1.4,1.7)	(1.9,1.7,1.3)
7	(1.4,1.6,1.8)	(1.7,1.4,1.2)	(1.7,1.3,1.9)	(1.3,1.6,1.8)	(1.4,1.7,1.2)	(1.2,1.9,1.1)	(1.8,1.6,1.2)	∞	(1.8,1.3,1.6)	(1.5,1.8,1.4)
8	(1.6,1.3,1.8)	(1.4,1.7,1.1)	(1.6,1.8,1.3)	(1.1,1.4,1.2)	(1.8,1.6,1.3)	(1.9,1.8,1.2)	(1.6,1.7,1.4)	(1.8,1.6,1.4)	∞	(1.7,1.3,1.6)
9	(1.4,1.2,1.6)	(1.8,1.3,1.7)	(1.6,1.8,1.2)	(1.7,1.42,1.12)	(1.6,1.4,1.3)	(1.3,1.4,1.1)	(1.7,1.8,1.5)	(1.6,1.9,1.4)	(1.6,1.8,1.3)	∞

Table 14 Input data: traveling time of traveling salesman vehicle

i/j	Crisp traveling time matrix(10×10) with three conveyances									
	0	1	2	3	4	5	6	7	8	9
0	∞	(18,12,13)	(16,19,14)	(18,17,20)	(11,15,10)	(10,28,23)	(21,23,24)	(21,14,19)	(22,24,20)	(17,16,18)
1	(21,20,23)	∞	(19,12,11)	(16,18,14)	(19,17,35)	(24,26,25)	(17,19,16)	(26,25,24)	(16,18,15)	(21,18,17)
2	(24,23,25)	(15,17,16)	∞	(16,19,17)	(15,17,18)	(18,19,21)	(32,36,38)	(16,17,19)	(22,24,26)	(16,17,19)
3	(16,15,18)	(15,18,17)	(33,32,45)	∞	(12,9,10)	(14,13,12)	(17,18,15)	(19,20,21)	(11,14,12)	(21,19,23)
4	(19,17,20)	(11,12,10)	(21,23,20)	(16,17,19)	∞	(12,10,14)	(13,14,15)	(16,18,19)	(18,22,24)	(16,18,20)
5	(13,14,16)	(18,19,22)	(13,12,14)	(30,29,28)	(14,16,18)	∞	(16,18,17)	(21,23,22)	(19,21,17)	(17,16,14)
6	(16,19,17)	(13,15,14)	(16,17,19)	(12,14,16)	(20,21,23)	(23,22,24)	∞	(16,14,41)	(16,18,14)	(26,29,32)
7	(11,10,9)	(15,20,21)	(18,19,17)	(24,22,21)	(19,18,21)	(15,14,16)	(24,26,28)	∞	(11,14,12)	(20,19,22)
8	(17,19,15)	(23,22,25)	(18,17,20)	(14,12,13)	(10,12,14)	(13,15,17)	(21,19,22)	(26,17,19)	∞	(14,16,15)
9	(15,16,14)	(16,18,17)	(16,14,17)	(10,12,15)	(20,25,26)	(18,17,19)	(21,20,23)	(23,22,25)	(17,16,19)	∞

Table 15 Input data: demand/requirement of every node

i/j	Crisp demand matrix (1 × 10)									
	0	1	2	3	4	5	6	7	8	9
	0	15	9	7	11	14	6	14	8	10

Table 16 Input data: unload time and cost of every node

i/j	Crisp unload time and cost matrix									
	0	1	2	3	4	5	6	7	8	9
Unload time	0	1.5	0.9	0.7	1.1	1.4	0.6	1.4	0.8	1.0
Unload cost	0	3.0	1.8	1.4	2.2	1.8	1.2	2.8	1.6	2.0

Table 17 Input data: servicing time and cost of every node

i/j	Crisp servicing time and cost matrix									
	0	1	2	3	4	5	6	7	8	9
Servicing time	0	3.0	1.8	1.4	2.2	1.8	1.2	2.8	1.6	2.0
Servicing cost	0	9.0	5.4	4.2	6.6	5.4	3.6	8.4	4.8	6.0

Table 18 Input data: distance matrix for goods vehicle for M/S Sharma Furniture company (in km.)

i/j	Kharagpur(0)	Midnapur(1)	Debra(2)	Sabang(3)	Salbani(4)	Jhargram(5)	Datan(6)	Gopiballabpur(7)	Binpur(8)	Keshpur(9)
Kharagpur(0)	∞	15.68	32.16	41.06	35.60	26.10	47.21	43.19	32.88	37.24
Midnapur(1)	15.68	∞	22.08	37.24	24.89	33.64	58.99	55.54	37.08	21.81
Debra(2)	32.16	22.08	∞	21.90	37.07	58.48	58.97	74.19	59.50	23.79
Sabang(3)	41.06	37.24	21.90	∞	58.57	67.03	45.76	79.55	73.68	44.65
Salbani(4)	35.60	24.89	37.07	58.57	∞	41.71	81.62	70.35	32.10	18.35
Jhargram(5)	26.10	33.64	58.48	67.03	41.71	∞	15.91	31.57	15.66	50.18
Datan(6)	47.21	58.99	58.97	45.76	81.62	15.91	∞	57.33	78.32	72.95
Gopiballabpur(7)	43.19	55.54	74.19	79.55	70.35	31.57	57.33	∞	45.40	77.48
Binpur(8)	32.88	37.08	59.50	73.78	32.10	15.66	78.32	45.40	∞	45.64
Keshpur(9)	37.24	21.81	23.79	44.65	18.35	50.18	72.95	77.48	45.64	∞

Table 19 Demand/requirement of every node for M/S Sharma Furniture company

i/j	Crisp demand matrix(1×10)									
	0	1	2	3	4	5	6	7	8	9
	0	19	12	10	14	16	12	18	9	15

Table 20 Unload time and cost of every node for M/S Sharma Furniture company

i/j	Crisp unload time and cost matrix									
	0	1	2	3	4	5	6	7	8	9
Unload time	0	1.8	1.1	0.9	1.2	1.8	1.6	1.4	0.8	1.4
Unload cost	0	3.5	2.8	1.7	2.5	3.8	3.2	2.5	1.9	2.7

Table 21 Servicing time and cost of every node for M/S Sharma Furniture company

i/j	Crisp servicing time and cost matrix									
	0	1	2	3	4	5	6	7	8	9
Servicing time	0	3.5	2.8	1.7	2.9	2.8	2.2	2.6	2.3	2.4
Servicing cost	0	14.0	11.2	6.8	11.6	11.2	8.8	10.4	9.2	9.6

Table 22 Parameter table of STSPwGDS

Parameter	Values
Total demand ($\sum_{i=1}^{N} d_i = D$)	94 kg
Transportation of goods cost ($\sum_{i=0}^{N-1}(D - \sum d_i)\xi$)	(Remaining total demand)(1/10) $
Goods unload time ($d_i t_2$)	(Demand of ith node)(1/10) h
Goods unload cost ($d_i w_4$)	(Demand of ith node)(2/10) $
Goods vehicle holding cost ($\Pi(i) = \|\theta_i - \phi_i\|\eta_2$)	(Time gap in hour)4 $
Day time halt limit (Ω_2)	5 h
Day time halt for goods vehicle ($\Pi(i) = \|\theta_i - \phi_i\|\eta_4 + \eta_5$)	(Time gap in hour)(1) + η_5 h
Some fixed charge for day time halting (η_5)	(Demand of ith node)(1/20) $
Traveling salesman stay cost ($\Pi(i) = \|\theta_i - \phi_i\|\eta_1$)	(Time gap in hour)6 $
Servicing time ($d_i t_1$)	(Demand of ith node)(2/10) h
Servicing cost ($d_i w_3$)	(Demand of ith node)(2/10) $
Servicing cost by third-party ($d_i w_6$)	(demand of ith node)w_6 $, w_6 is bargaining parameter
(Consider) servicing cost by third-party ($d_i w_6$)	(Demand of ith node)(60/10) $
(Consider) servicing time by third-party ($d_i t_1$)	(Demand of ith node)(2/10) h

References

Albayrak M, Allahverdi N (2011) Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms. Expert Syst Appl 38(3):1313–1320

Averbakh I, Yu W (2018) Multi-depot traveling salesmen location problems on networks with special structure. Ann Oper Res 635–648

Bailay R (2018) Ikea bets on small stores to go digital. https://economictimes.indiatimes.com/industry/services/retail/ikea-bets-on-small-stores-to-go-digital/articleshow/66743638.cms?from=mdr

Baldwin JM (1896) A new factor in evolution. Am Nat 30(354):441–451

BBC (2018) Surrogate mother of 'twins' finds one is hers. https://www.bbc.com/news/health-41858232

Beamurgia M, Basagoiti R, Rodríguez I, Rodríguez V (2022) Improving waiting time and energy consumption performance of a bi-objective genetic algorithm embedded in an elevator group control system through passenger flow estimation. Soft Comput 26(24):13673–13692

Carrabs F, Cerulli R, Sciomachen A (2017) An exact approach for the grocery delivery problem in urban areas. Soft Comput 21(9):2439–2450

Chang T-S, Wan Y-W, Ooi WT (2009) A stochastic dynamic traveling salesman problem with hard time windows. Eur J Oper Res 198(3):748–759

Constantinescu E (2012) Three dimensional mari time transportation models. https://www.scribd.com/document/419467025/Three-Dimensional-Maritime-Transportation-Models

Cordeau J-F, Laporte G, Potvin J-Y, Savelsbergh MW (2007) Transportation on demand. Handbk Oper Res Manag Sci 14:429–466

Corporation X (2012) Reliable installation service from MI. https://www.mi.com/in/service/tv_installationservice/

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

Dong G, Guo WW, Tickle K (2012) Solving the traveling salesman problem using cooperative genetic ant systems. Expert Syst Appl 39(5):5006–5011

Eiben AE, Raue PE, Ruttkay Z (1994) Genetic algorithms with multi-parent recombination. In: International conference on parallel problem solving from nature. Springer, pp 78–87

Feng Z (2019) Constructing rural e-commerce logistics model based on ant colony algorithm and artificial intelligence method. Soft Comput 23(15):6697–6714

Focacci F, Lodi A, Milano M (2002) A hybrid exact algorithm for the TSPTW. INFORMS J Comput 14(4):403–417

Forum WE (2018) Where people work the longest hours. https://www.weforum.org/agenda/2018/01/the-countries-where-people-work-the-longest-hours/

Ghoseiri K, Sarhadi H (2008) A memetic algorithm for symmetric traveling salesman problem. Int J Manag Sci Eng Manag 3(4):275–283

Haley K (1963) The multi-index problem. Oper Res 11(3):368–379

Haxhimusa Y, Carpenter E, Catrambone J, Foldes D, Stefanov E, Arns L, Pizlo Z (2011) 2d and 3d traveling salesman problem. J Problem Solv 3(2):8

Lagarteja JG, Gerardo BD, Medina RP (2017) Improved genetic algorithm using new crossover operator. In: Proceedings of 4th international conference on innovations in engineering, technology, computers and industrial applications (IETCIA-17) Aug, pp 3–4

Lawler E, Lenstra J, Rinnooy Kan A, Shmoys D (1985) The traveling salesman problem: Ge re guided tour of combinatorial optimization. Wiley and Sons, New York

Lee CW, Wong WP (2022) Last-mile drone delivery combinatorial double auction model using multi-objective evolutionary algorithms. Soft Comput 26:12355–12384

Ma Y, Li Z, Yan F, Feng C (2019) A hybrid priority-based genetic algorithm for simultaneous pickup and delivery problems in reverse logistics with time windows and multiple decision-makers. Soft Comput 23(15):6697–6714

Maity S, Roy A, Maiti M (2015) A modified genetic algorithm for solving uncertain constrained solid travelling salesman problems. Comput Ind Eng 83:273–296

Maity S, Roy A, Maiti M (2017) An intelligent hybrid algorithm for 4-dimensional tsp. J Ind Inf Integr 5:39–50

Majumdar J, Bhunia AK (2011) Genetic algorithm for asymmetric traveling salesman problem with imprecise travel times. J Comput Appl Math 235(9):3063–3078

Malaguti E, Martello S, Santini A (2018) The traveling salesman problem with pickups, deliveries, and draft limits. Omega 74:50–58

Martínez-Estudillo AC, Hervás-Martínez C, Martínez-Estudillo FJ, García-Pedrajas N (2005) Hybridization of evolutionary algorithms and local search by means of a clustering method. IEEE Trans Syst Man Cybern B (Cybern) 36(3):534–545

Merz P, Freisleben B (2001) Memetic algorithms for the traveling salesman problem. Complex Syst 13(4):297–346

Moon C, Kim J, Choi G, Seo Y (2002) An efficient genetic algorithm for the traveling salesman problem with precedence constraints. Eur J Oper Res 140(3):606–617

Moscato P et al (1989) On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826:1989

Nagata Y, Soler D (2012) A new genetic algorithm for the asymmetric traveling salesman problem. Expert Syst Appl 39(10):8947–8953

Petersen HL, Madsen OB (2009) The double travelling salesman problem with multiple stacks-formulation and heuristic solution approaches. Eur J Oper Res 198(1):139–147

Reardon S (2017) Genetic details of controversial 'three-parent baby' revealed. Reprod Biol 544:17–18

Reinelt G (1995) Tsplib. http://www.iwr.uni-heidelberg.de/groups/comopt/software. TSPLIB95

Rodriguez-Roman D (2018) A surrogate-assisted genetic algorithm for the selection and design of highway safety and travel time improvement projects. Saf Sci 103:305–315

Roy A, Chakraborty G, Khan I, Maity S, Maiti M (2016) A hybrid heuristic for restricted 4-dimensional TSP (R-4DTSP). In: International conference on frontiers in optimization: theory and applications. Springer, pp 285–302

Sarkar R (2014) Ro leads, but kent thirsty for more

Skinner MK (2015) Environmental epigenetics and a unified theory of the molecular aspects of evolution: a neo-Lamarckian concept that facilitates neo-darwinian evolution. Genome Biol Evol 7(5):1296–1302

Tüű-Szabó B, Földesi P, Kóczy LT (2017) An efficient new memetic method for the traveling salesman problem with time windows. In: International workshop on multi-disciplinary trends in artificial intelligence. Springer, pp 426–436

Wang J, Ersoy OK, He M, Wang F (2016) Multi-offspring genetic algorithm and its application to the traveling salesman problem. Appl Soft Comput 43:415–423

Wang Y (2015) An approximate method to compute a sparse graph for traveling salesman problem. Expert Syst Appl 42(12):5150–5162

Wang Y, Li J, Pan Q, Sun J, Ren L (2010) Memetic algorithm based on improved inver-over operator for TSP. In: 2010 sixth international conference on natural computation (ICNC), vol 5. IEEE, pp 2386–2389

Wang Z, Lin W-H (2017) Incorporating travel time uncertainty into the design of service regions for delivery/pickup problems with time windows. Expert Syst Appl 72:207–220

Xu X, Yuan H, Matthew P, Ray J, Bagdasar O, Trovati M (2019) Gorts: genetic algorithm based on one-by-one revision of two sides for dynamic travelling salesman problems. Soft Comput 24:7197–7210

Ye T, Wang T, Lu Z, Hao JK (2014) A multi-parent memetic algorithm for the linear ordering problem. arXiv preprint arXiv:1405.4507