Indian Institute of Management Calcutta

Working Paper Series

WPS No 848 /August, 2020

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

Somnath Majhi

Dept. of Computer Science and Engineering,Maulana Abul Kalam Azad University of Technology,
Nadia-741249, W.B., India

Samir Maity

Operations Management Group, IIM CalcuttaCalcutta
Joka, D H Road, Kolkata-700104, India

Sumanta Basu*

Professor, Operations Management Group, IIM Calcutta
Phone: 91-9051054433, Email: sumanta@iimcal.ac.in

Debashis Giri

Professor, Department of Information Technology, Maulana Abul Kalam Azad University of Technology
Nadia-741249, W.B., India

Manoranjan Maiti

(Former Professor)

Department of Applied Mathematics With Oceanology and Computer Programming
Vidyasagar University, Midnapur-721102, India

***Corresponding Author**

Indian Institute of Management Calcutta, Joka, D.H. Road, Kolkata 700104

URL: http://facultylive.iimcal.ac.in/workingpaper

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

S. Maji[1] · **S. Maity**[2] · **S. Bsau**[2] · **D. Giri**[3] · **M. Maiti**[4]

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 above activities is essential to reduce customer dissatisfaction, to curb the various waiting or demurrage costs due to delayed arrivals of goods vehicles and service workers. This paper attempts to model and solve such a realistic synchronized goods delivery and service problem against online booking. In this model, one goods vehicle starts from the company's storehouse with all of the goods to be delivered and moves continually, dropping the goods at the specified locations. For service, a service worker separately moves and uses the appropriate conveyances among the available ones at each node to reach the customers. This paper poses an interesting research question to understand the requirements of separate tour path for goods vehicles and service worker along with appropriate conveyance for serviceman's arrival. This is a 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 are developed and tested on some standard test functions to establish its superiority over the standard ones. VOMA implementation on above proposed problem reveals the influence of unloading and service times, halt time availability and time windows, and third-party outsourcing charges on final 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 Salesperson Problem and Servicing; Three-parent Crossover;

S. Maji
E-mail: somnathmajivucs@gmail.com

S. Maity
E-mail: samirm@iimcal.ac.in
· S. Basu
E-mail: sumanta@iimcal.ac.in
· D. Giri
E-mail: debasis_giri@hotmail.com
· M. Maiti
E-mail: mmaiti2005@yahoo.co.in
· [1] Dept. of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, Nadia-741249, W.B., India
· [2] Operations Management Group, Indian Institute of Management Calcutta, Joka, D H Road, Kolkata-700104, India
· [3] Department of Information Technology, Maulana Abul Kalam Azad University of Technology, Nadia-741249, W.B., India
· [4] Department of Applied Mathematics With Oceanology and Computer Programming, Vidyasagar University, Midnapur-721102

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

1 Introduction

A traveling salesman problem (TSP) aims to determine the round-trip tour for a 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 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., company’s service person) 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 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 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 an objective of minimizing the total cost, 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 service worker travel and goods delivery with service. In the following sections, we refer to this problem as ’solid TSP with goods and services delivery’ (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 service person 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 service people 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 service worker 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 service people and goods vehicle so that the overall cost for the system is minimal.

We consider a single vehicle type for 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 the addition of one more dimension, i.e., vehicle type. Following this model, service worker 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 the expansion of e-commerce models is dependent on geographic reach. Our third research question attempts to understand the implications of vehicle type in designing the routes for service worker 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 sensitivity of decision making to parameters from the problem context.

STSPwGDS is a *NP*-hard problem. Considering *NP*-hard nature of the problem, our fifth research contribution is in developing 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 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 Friedmans 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 paper is organized as follows. Section 1 presents a brief introduction of the problem context and the heuristics developed, followed by section 2, which provides a literature review. The mathematical description of the problem is elaborated in section 3. Section 4 explains the VOMA heuristic, with Section 5.3.1 providing details about computational experiments to establish the efficiency of the VOMA through some statistical tests. Sections 6 and 6.2.1 conduct performance analysis of the heuristic and provide a follow-up discussion. Practical implementation and the managerial insights obtained by implementing the heuristic are elaborated in Sections 7 and 8, respectively. We conclude the paper by summarizing the major contributions and indicating future research avenues in Section 9.

2 Literature review

The TSP is a combinatorial NP-hard optimization problem [Lawler et al. (1985)]. 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.

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 constraint, 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, vegetable agricultural products solved through ant colony algorithm. The main focus was on difficulties at 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. This literature gap relates to the first, second and fourth research questions addressed in this paper.

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 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, as considered in our third research question.

To solve these *NP*-hard combinatorial optimization problems 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. 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.

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 2-opt and 3-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.

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 3.1 presents standard TSP formulation. Section 3.2 extends it by considering multiple vehicle types. In section 3.3, we formulate the proposed STSPwGDs problem by including goods delivery and service within TSPs.

3.1 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 ith city to jth city. The mathematical model is as follows:

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

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

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

along with the sub-tour elimination equation 2.

3.2 Solid Traveling Salesman Problem or 3-Dimensional TSP (3D TSP)

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

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

along with the sub-tour elimination equation 2.

3.3 Solid Traveling Salesman Problem with Goods Delivery and Service (STSPwGDS)

This section addresses the first and third research questions as mentioned in section 1. Here an exclusive car with goods starts from the retailer's godown/warehouse and returns to it dropping the appropriate article(s) at customer's locations (nodes) as the demand placed online. At the same time, service man also starts from the godown/warehouse and returns to it after giving services 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 serviceman at a station, the waiting charge for the serviceman 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 serviceman so that total cost of the system (STSPwGDS), including the unloading and service charges, is minimal. Following the second research question, we develop two different models to understand the differences in the solutions obtained.

Model 1: Goods vehicle and service worker follow optimal travel routes to minimize total costs.

Model 2: Service worker follows the optimal travel route of the goods vehicle or vice-versa.

3.3.1 Mathematical formulations of Models 1 and 2

Following equation (4), the problem is mathematically formulated as:

$$\left.\begin{aligned} &\text{Minimize} \quad Z = \sum_{i=0}^{N-1} \alpha(x_i, x_{i+1}, v_p) + \alpha(x_N, x_0, v_l) + \sum_{i=0}^{N-1} \beta(y_i, y_{i+1}, V) + \beta(y_N, y_0, V) \\ &\qquad + \sum_{i=0}^{N-1} (D - \sum d_i)\xi + \sum_{i=1}^{N} \omega(d_i) + \sum_{i=1}^{N} \Pi(i) + \sum_{i=1}^{N} \gamma(d_i) \\ &\qquad\qquad \text{subject to } \sum_{i=1}^{N} d_i = D \\ &\qquad\qquad \textit{where } \; p, l \in \{1, 2, \cdots, q\} \\ &\qquad\qquad \omega(d_i) = d_i * w_4 \\ &\qquad\qquad \gamma(d_i) = d_i * w_3 \end{aligned}\right\} \tag{5}$$

D represents the total demand throughout the process, and d_i indicates the demand for the ith node/station.

Here, $\alpha(x_i, x_{i+1}, v_p)$ indicates the travel cost for the service worker from the i^{th} to $(i+1)^{th}$ nodes using the p^{th} vehicle, and $\beta(y_i, y_{i+1}, V)$ indicate the transportation cost of the goods vehicle between the i^{th} node and the $(i+1)^{th}$ node using a particular vehicle V. $\sum_{i=0}^{N-1} (D - \sum d_i)\xi$ represents exclusively the transportation cost of goods, which depends on the transported amount between nodes, and ξ indicates the cost per unit distance per unit weight. $\omega(d_i) = d_i * w_4$ represents the unloading cost at the i^{th} node, w_4 is the unloading charge per unit weight.

The penalty cost expression $\Pi(x_i)$ is represented as:

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

Penalty weightages η_1 and η_2 are used for corresponding service worker and goods vehicle penalties per unit of time. Again, θ_i and ϕ_i indicate the cumulative time taken by the service worker and goods at the i^{th} node.

$\gamma(d_i)$ $(= d_i * w_3)$ represents the servicing cost at the ith node, w_3 is the service charge per unit of demand.

Model 2 makes minor modifications to the objective function only, and the revised objective function is as follows:

$$\left.\begin{aligned} &\text{Minimize} \quad Z = \sum_{i=0}^{N-1} \alpha(x_i, x_{i+1}, v_p) + \alpha(x_N, x_0, v_l) + \sum_{i=0}^{N-1} \beta(x_i, x_{i+1}, V) + \beta(x_N, x_0, V) \\ &\qquad + \sum_{i=0}^{N-1} (D - \sum d_i)\xi + \sum_{i=1}^{N} \omega(d_i) + \sum_{i=1}^{N} \Pi(i) + \sum_{i=1}^{N} \gamma(d_i) \end{aligned}\right\} \tag{6}$$

3.3.2 Models 1 and 2 with waiting time restrictions

This section develops the modified formulation to address the fourth research question on the impact of allowable halting times. Following the proposed model, the goods vehicle can wait at designated locations to avoid the penalty of reaching the customer location early. We present a variation by considering constraints on waiting times in some cases, instead of demurrage/stay charges.

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

Let the starting time be 6 A.M.

current time= (Cumulative time) mod 24

if (current time < 12) and $(\theta_i - \phi_i < \Omega_2)$

$$\Pi(i) = \begin{cases} |\theta_i - \phi_i| * \eta_4 + \eta_5 & : \theta_i - \phi_i < \Omega_2 \\ |\theta_i - \phi_i| * \eta_2 & : \theta_i < \phi_i \\ 0 & : \theta_i = \phi_i \end{cases}$$

else

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

Penalty weightages η_1, η_4 and η_2 are used for corresponding service person and goods vehicle penalties per unit of time. Additionally, η_5 is used for some fixed charge for every halt. Ω_2 is the maximum daytime halt of the goods vehicle.

Model-2(a): Model 2 with halting of the goods' vehicle at some nodes following a time constraint.

Equation 6 from Model-2, with the penalty condition of Model-1(a) added.

3.3.3 STSPwGDS through third-party service (Model 3)

Continuing with the fourth research question, we understand the applicability of the models developed in the presence of a third-party service provider and the fees charged by them. When outsourcing the service, the company assigns it to a third party, instead of its own service worker. For this work, a charge is paid to the third party. In this consideration, there will be no need for service workers and no demurrage or stay charges. The mathematical formulation of the model is as follows.

$$\left.\begin{array}{c} \text{Minimize} \quad Z = \sum_{i=0}^{N-1} \beta(y_i, y_{i+1}, V) + \beta(y_N, y_0, V) \\ + \sum_{i=0}^{N-1} (D - \sum d_i)\xi + \sum_{i=1}^{N} \omega(d_i) + \sum_{i=1}^{N} (d_i * w_6) \\ \text{subject to} \ \sum_{i=1}^{N} d_i = D \\ \omega(d_i) = d_i * w_4 \end{array}\right\} \tag{7}$$

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

Following the fifth research question, this section elaborates on incorporating varied offspring concepts into our proposed heuristic. The motivation came from natural fertilization processes, in which a sperm fuses with one or more eggs randomly to create a number of offspring, i.e., singletons, twins, triplets, quadruplets, etc. Currently, in gestational surrogacy, these embryos are placed in the womb of another surrogate mother, and offspring are born through the involvement of three parents. It is said that the nature, behavior, mental condition, etc., of the child are influenced by the surrogate mother (cf. BBC (2018)). Moreover, recently the '3-parent' concept (cf. Reardon (2017)) has been used to combine the DNA from 3 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}, $\cdots$, x_{iN}), where x_{i1}, x_{i2}, $\cdots$, x_{iN} indicate N consecutive nodes in a tour. A population with size M is defined by x_{ij}, i=1, 2, $\cdots$, M and j=1, 2, $\cdots$, 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 and Crossover Operator

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, $p_B = e^{((g/G)*(f_{min}-f(X_i))/KT)}$, where $T = T_0(1-a)^k$, $k = (1 + C * rand[0,1])$, C=rand[1, 100], g=current generation number, G= maximum generation, T_0= rand[60, 150], a=rand[0, 1], and $f(X_i)$ is the objective function.
To form the mating pool, a predefined value, for instance, 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 $r < p_s$, then the corresponding chromosome is stored in the mating pool; otherwise, the corresponding chromosome is selected for the mating pool, and the chromosome according to f_{min} is taken to the mating pool. Algorithm 1 describes the steps involved.

Algorithm 1 Modified probabilistic selection procedure

Require: Max-gen (G), Probability of selection (p_s), pop - size (M).
Ensure: Mating pool.
Step 1: start algorithm
Step 2: for (n=1 to M)
Step 3: r= random value between o and 1;
Step 4: T_0= random value between 60 and 150;
Step 5: a=rand[0, 1] and C=rand[0, 100];
Step 6: k=(1+C*rand[0, 1]);
Step 7: $T=T_0(1-a)^k$;
Step 8: $p_B = e^{((g/G)*(f_{min}-f(X_i))/KT)}$;
Step 9: if ($r < p_s$)
Step 10: choose the corresponding chromosome;
Step 11: else if ($r < p_B$)
Step 12: choose the corresponding chromosome;
Step 13: else
Step 14: select the chromosome corresponding to f_{min};
Step 15: end for;
Step 16: end for;
Step 16: create matting pool
Step 18: end algorithm

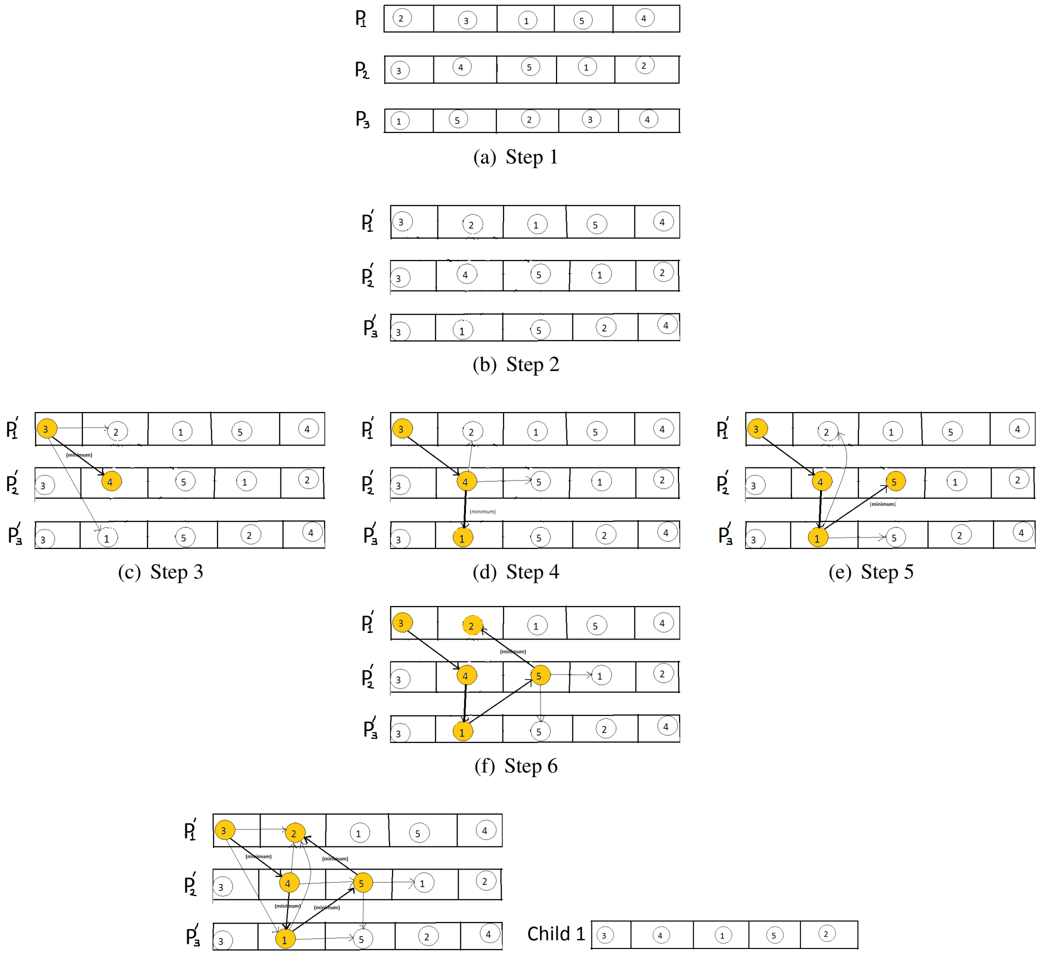
4.2.2 Determination of Varying Numbers of Offspring Surro-Embryos crossover

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 introduce 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.

4.2.3 Selection of parents and Operationalization of Surro-Embryos crossover

We chose the number of parents for crossover using the following formula.
Number of Parents (NOP) = (p_c)*(Number of total population), where p_c is the probability of crossover (given). From the mating pool, we select randomly three parents and create a “parent group” for crossover and continue this formation of groups until eligible parents are available. Again, depending on the Random Offspring Number (RON), we produce a different number of offspring with an upper limit of 4 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 Figure 1.

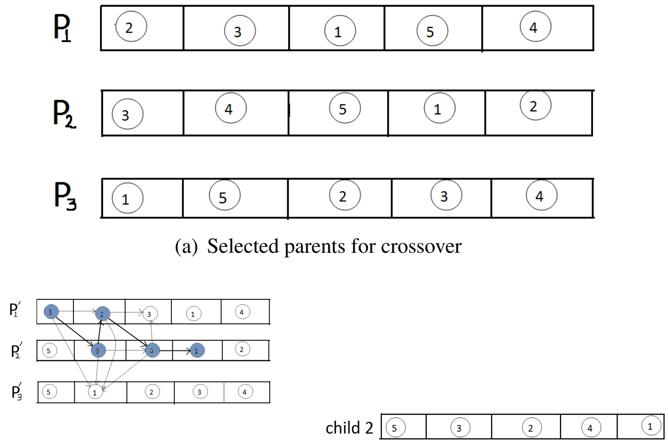


(a) Step 1
(b) Step 2
(c) Step 3
(d) Step 4
(e) Step 5
(f) Step 6
(g) All steps at a glance
(h) Child 1 produce

Fig. 1: Surro-Embryos Crossover for child 1

The three chosen parents are $P_1 = \{2\ 3\ 1\ 5\ 4\}$, $P_2 = \{3\ 4\ 5\ 1\ 2\}$, $P_3 = \{1\ 5\ 2\ 3\ 4\}$. 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 $P_1^{'} = \{3\ 2\ 1\ 5\ 4\}$, $P_2^{'} = \{3\ 4\ 5\ 1\ 2\}$, $P_3^{'} = \{3\ 1\ 5\ 2\ 4\}$.
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 Figure 1), child 2 (shown in Figure 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 section 2 with a flowchart in Figure 3.



(a) Selected parents for crossover

(b) Crossover for child 2

(c) After crossover produce child 2

Fig. 2: Surro-Embryos Crossover for child 2

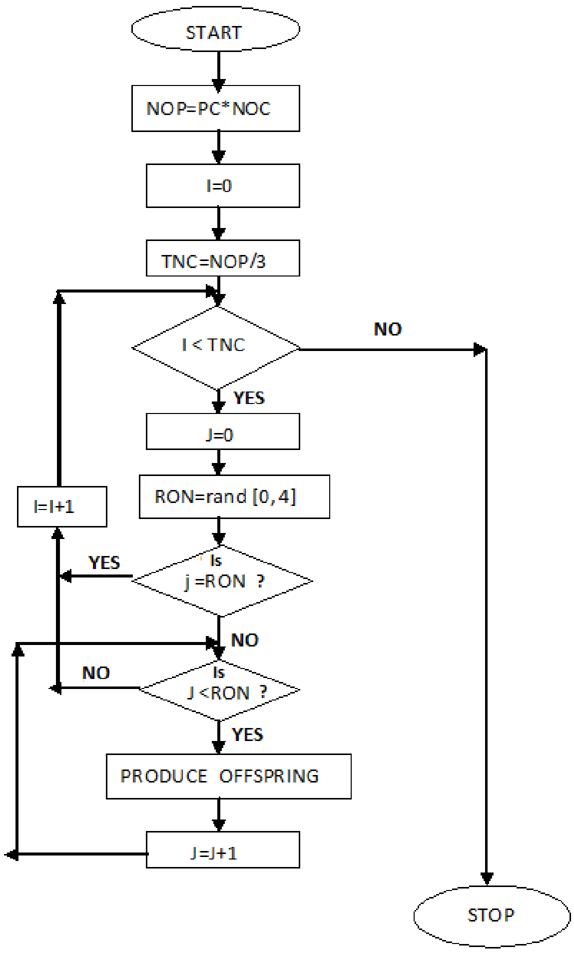


Fig. 3: Flowchart of Surro-Embryos crossover

4.3 Mutation

4.3.1 Fibonacci Generation Dependent mutation (Fibo-GD)

This section elaborates on our modified mutation mechanism. It is expected that values of p_m will decrease with the increase in generation. Here, we develop a generation-dependent mutation probability p_m using the

Algorithm 2 Algorithm of proposed Surro-Embryos crossover

Require: Write
Ensure: Write
Step 1: start algorithm
Step 2: number of parents (NOP) selected for mating pool = (p_c)*(pop-size).
Step 3: total number of crossovers (TNC) will be = NOP/3.
Step 4: **for**(i=1; i< TNC; i++)
Step 5: choose randomly three distinct parents from mating pool.
Step 6: generate random offspring number (RON)=rand[0, 4].
Step 7: initialize randomly three-parent (P_1, P_2, P_3) depending on the probability of crossover p_c.
Step 8: **for** (j=0; i< RON; j++)
Step 9: generate random number r_1=[0, node] (a_i say).
Step 10: place a_i as the first node of the child's chromosome.
Step 11: a_i=min(a_i, each next node) (say).
Step 12: find the minimum costs between a_i and each subsequent node of the given parents. (s_1 say).
Step 13: replace by s_1 in the next position of the child.
Step 14: repeat steps 11 and 13 until the end of the nodes.
Step 15: end for
Step 16: end for
Step 17: end algorithm

concept of Fibonacci function generating monotonic increasing series, with the mutation probability using the inverse of Fibonacci function being smooth and decreasing monotonically with generation. Thus, p_m is defined as

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

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. For the first generation, we consider $p_m=1$ and follow the generation-dependent Fibonacci series for subsequent generations.

4.3.2 Mutation process

Now, for a particular node-dependent problem like TSPs, to mutate a chromosome $X = (x_1, x_2, \cdots, x_N)$, $(v_1, v_2, \cdots, v_P)$. 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 provides a stepwise description of the mutation process.

Algorithm 3 Algorithm for Fibo-Generation-dependent random mutation

Require: Write
Ensure: Write
1: **start algorithm**
2: set g=current generation number
3: if (g = 1)
4: p_m=0.2.
5: else
6: $p_m = \frac{k}{\sqrt{f_g}}$, k∈(0, 1).
7: **for**(i=0; i< pop_size; i++)
8: r=rand(0, 1)
9: **if**(r< p_m)
10: select current chromosome
11: a=rand[1, N]
12: b=rand[1, N]
13: **if** (a==b) Goto step 8
14: **for** (j=1; j<=N; j++) //N=number of nodes in a path.
15: **if** (x[j]==a)
16: p=j;
17: **if** (x[j]==b)
18: q=j;
19: x[p]=b; x[q]=a; // replace a with b and b with a.
20: **end for**
21: end if
22: **end for**
23: **end algorithm**

Algorithm 4 Algorithm for proposed VOMA

Require: 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).
Ensure: optimum solutions.

```
start
g ← 0 // g: iteration/generation number
initialize //according to subsection 4.1
computefitness //according to subsection 4.1
while (g ≤ max_gen) {
selection operation
for every chromosome {
determine the probability of each chromosome according to sub-section 4.2.1
mating pool produce according to Algorithm 1
}
// crossover operation according to section 4.2.3
select three parents for crossover using p_c from the mating pool
for three-parent {
modify the parents;
generate offspring according to Algorithm 2
}
// mutation
generate p_m according to that given in section 4.3.1
select the offspring for mutations based on p_m
for selected chromosome {
swap the nodes according to Algorithm 3
}
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 optimal solutions.
g ← g+1
} //endwhile
store the global optimal results;
end algorithm.
```

4.4 VOMA algorithm and complexity analysis

A complete stepwise description is provided in Algorithm 4 that summarizes our methodological contribution while answering the fifth research question. We evaluate the time and space complexity of the algorithm below.

4.4.1 Time complexity

The complexity of GA for TSPs depends on the generation number (G), pop size (M), and numbers of nodes (N). The time complexity of SGA is O(GMN). Due to the comparison crossover mechanism (within Surro-Embryos crossover) the proposed VOMA has O(GMN^2) computational complexity.

4.4.2 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 to address the last research question. We move to answer the remaining questions in section 6 once the efficiency of the methodology is established.

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 1, to test the robustness. Based on the results obtained, we fix p_c as 0.3 and maximum number of generations as 2500 for VOMA implementation.

Table 1: 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	-	

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 2 reports the results of said classical TSP starting from 16 nodes to a maximum of 783 nodes by VOMA and SGA-1. These results are compared with respect to total cost, iterations and CPU time in seconds.

Table 2: Results for standard TSP problems (TSPLIB)

Instances	Best known solution (BKS)	VOMA			SGA-1		
		Cost (Rs.)	Iteration	Time (sec.)	Cost (Rs.)	Iteration	Time (sec.)
us16	6859	6859	**47**	0.03	6859	242	0.04
gr17	2085	2085	**64**	0.05	2085	353	0.10
gr21	2707	2707	**148**	0.09	2707	418	0.12
bays29	2020	2020	**132**	0.14	2020	397	0.42
eil51	426	426	**396**	0.51	426	514	1.62
st70	675	675	**341**	0.73	810	589	1.83
eil76	538	538	**578**	0.74	675	635	2.03
eil101	629	703	**764**	1.65	715	926	2.32
kroA150	26524	27105	**873**	1.75	28396	1098	3.02
kroB150	26130	26958	**894**	1.87	29615	1104	3.18
a280	2579	3168	**1124**	2.66	3442	1338	4.47
lin318	42029	43764	**1235**	3.76	44859	1407	5.85
pcb442	50778	52947	**1407**	4.45	58107	1693	6.49
rat783	8806	9726	**1688**	6.19	11257	2119	7.63

VOMA outperformed Standard GA in terms of both solution quality obtained and computational time. The 'Iteration' column in Table 2 indicates the number of iterations that the heuristic requires to reach the best solution. We observe much faster convergence to the optimal 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 3 summarizes the best known solution (BKS) and average and standard deviation (SD) of results obtained from a particular heuristic over 8 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 benchmark 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 relative error in different cases. These errors are also very small, indicating that the derived average solutions are nearer to the BKS. Thus, the proposed VOMA has produced results closer to the optimal results. Next, we conduct statistical tests to understand the relative difference between the results obtained using VOMA and the other four GA-based algorithms.

Table 3: 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
SGA-I	Avg	7132.72	2208.16	2812.56	2143.24	595.62	1061.52	707.15	975.24
	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
SGA-II	Avg	6901.18	2107.82	2731.64	2129.75	514.63	896.28	697.14	912.43
	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
SGA-III	Avg	6876.92	2149.35	2774.52	2089.62	511.58	853.17	693.38	866.71
	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
SGA-IV	Avg	6984.32	2187.51	2793.18	2082.38	579.74	1039.36	698.79	949.63
	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
VOMA	Avg	6868.95	2093.24	2711.46	2022.16	445.12	729.81	618.63	711.54
	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

5.3.2 Friedmans test

To compare the performance of the algorithms SGA-I, SGA-II, SGA-III, SGA-IV, and VOMA, we perform Friedmans test (Derrac et al. (2011)). It is a non-parametric statistical procedure, the main aim of which is to detect a significant difference between the behavior of two or more algorithms.
The following assumptions of Friedmans test are:

- The results over instances (problems from TSPLIB) are mutually independent (i.e., the results within one instances does 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 4) is prepared based on average values as reported in Table 3.

Table 4: Ranking of Friedmans test

Algorithms(k)	SGA-I	SGA-II	SGA-III	SGA-IV	VOMA
Instances(b)	R(X_{b1})	R(X_{b2})	R(X_{b3})	R(X_{b4})	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

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, $\cdots$, 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 4, 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 α= 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 Friedmans 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 α=0.01 and 28 degrees of freedom is 2.76, and the critical value for the difference is = 17.74.
Table 5 summarizes the paired comparisons, and the underlined values indicate the extent of differences between the algorithms. From Table 5, we conclude that VOMA has outperformed all of the other algorithms.

Table 5: Paired comparison of Friedmans 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 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 service worker'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 are presented in Tables A1, A2, A3, A4, A5, A10, A11, A12 and A9, respectively. Data specific to the model parameters are reported in Table A13.

6.2 Optimum results of STSPwGDS under different Models

Table 6 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 Table 7 and Table 8.

Table 6: Model description

Cases	Conditions
Model-1	Individual optimal paths are determined for the goods vehicle and service worker by minimizing overall cost.
Model-2	Optimal travel path for goods vehicle is followed by service worker 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.

Table 7: Optimal results of Model-1

	Goods vehicle	path	0	5	1	9	4	8	3	2	6	7	Total
	Service worker vehicle	path	0(0)	5(1)	9(2)	1(2)	4(1)	8(1)	3(2)	2(2)	6(2)	7(2)	
	Transportation	time	24	20	47	39	41	26	45	43	40	30	355
		cost	48	240	32	88.05	72	199.92	56	72	80.04	159.9	1047.91
	Goods	time	-	-	-	-	-	-	-	-	-	-	-
		cost	-	9.40	8.00	6.50	5.50	4.40	3.60	2.90	2.00	1.40	43.70
Model-1	Unloading	time	-	1.4	1.5	1.0	1.1	0.8	0.7	0.9	0.6	1.4	9.4
		cost	-	2.80	3.00	2.00	2.20	1.60	1.40	1.80	1.20	2.80	18.80
	Traveling	time	10	17	17	35	22	12	45	38	41	9	246
		cost	13	54.4	54.4	36.96	19.2	39.2	11.12	12.10	13.20	54	307.58
	Holding/stay	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
		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
	Servicing	time	-	2.8	3	2	2.2	1.6	1.4	1.8	1.2	2.8	18.8
		cost	-	8.40	9.00	6.00	6.60	4.80	4.20	5.40	3.60	8.40	56.40
	Total	time											797.1
		cost											2311.39
	Total	Transportation distance	10	32	5	15	12	28	8	10	12	30	162
		Traveling distance	10	32	32	33	12	28	8	10	12	30	207

Figure 4 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 optimal travel tour. Although our results indicate Model 3 as our preferred choice, it is primarily driven by the outsourcing fee offered to the third party. A detailed discussion of the effect of outsourcing fee is provided while discussing Figure 7.

6.2.1 Discussion of the results

Following Tables 7 and 8, 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 service worker 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 service worker 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 hrs in both Models-1 and -2, then from Table 8, Model 1(a) yields a lesser total cost (\$2310.34) than Model-2(a) (\$2428.81).

Table 8: Optimal results of Models 2, 3, 1(a), and 2(a)

	Goods/Serviceman	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
Model-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	Transportation/Traveling distance	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
Model-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	Transportation/Traveling distance	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
	Service worker 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
Mode-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 (daytime halt)/0	0/3.8	.8(just over day time)/0	85.2/82.7
	Holding/stay	cost	-	0/92.40	268/298.20	0/0	67.60/0	0/4.2	0/78.60	(0.5+0.45)(day time halt)/0	0/22.80	3.20/0	(0.95)+338.8/496.20
		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
		Transportation distance	10	32	5	15	12	28	8	10	12	30	162
	Total	Traveling distance	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
Mode-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	Transportation/Traveling distance	35	10	12	6	5	15	12	23	10	26	154

Figure 4 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 8 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 hrs., 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 service worker's travel and stay 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 σ_{cost}=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 hrs (Model 3). Hence, the total time taken =(403.4+18.8)hrs=422.2 hrs, which is still less than with Model 1(a) because the total service charge is still less than the service worker'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 optimal results are obtained and are presented graphically in Figure 5. In both cases, the optimal value reduces with decreasing returns.

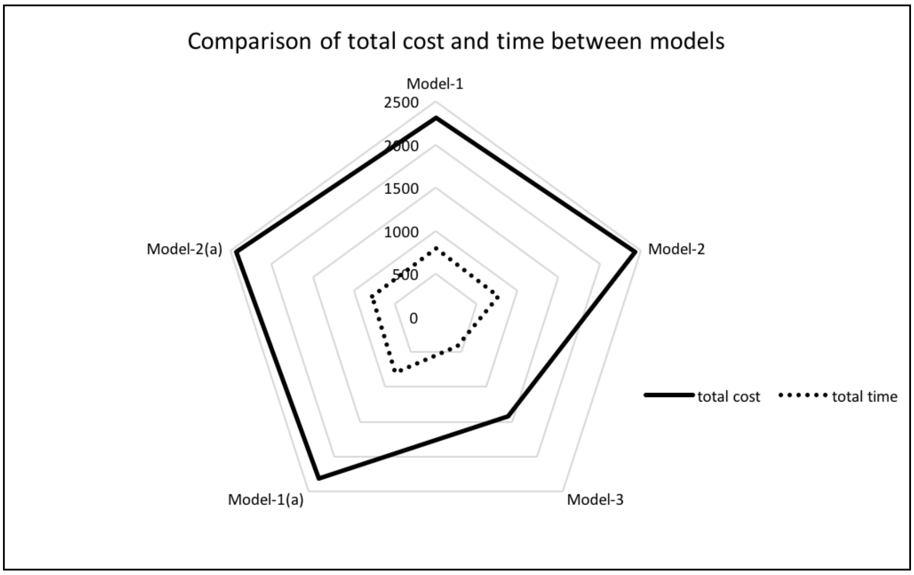
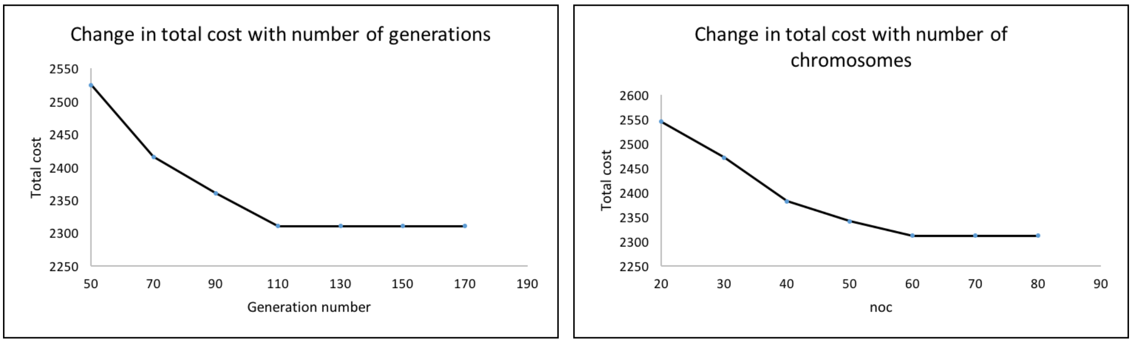


Fig. 4: Total cost and travel time of all models

Figures 5(a) and 5(b) illustrate the effects on total cost with generation number and number of chromosomes, respectively.



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

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

Fig. 5: 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 the 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 10 customer locations to understand the implications. Figure 6 shows separate routes for the goods vehicle and salesperson following Model 1 and the optimal route for the transport vehicle followed by the salesperson in Model 2.

We solve Model 1 using VOMA to obtain the total optimal 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

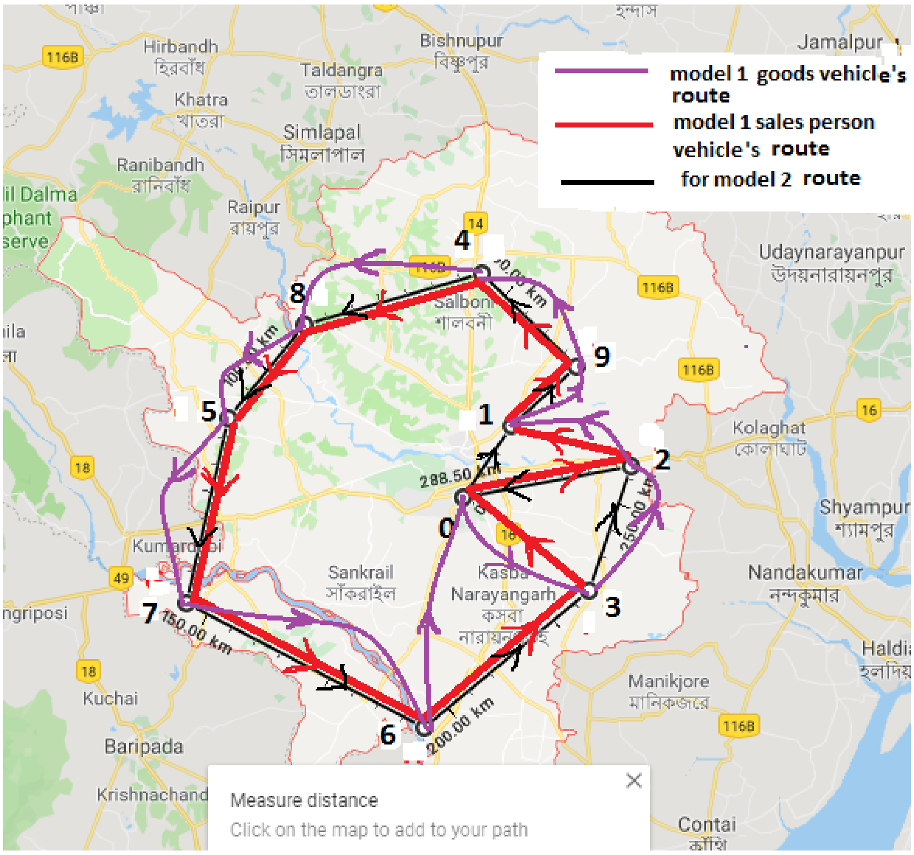


Fig. 6: Optimal paths using Models 1 and 2 for the Shrama Furniture company

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 optimal cost for Model-1 ($3493.56) is less than overall optimal cost for Model 2 ($3587.48).

8 Managerial insight

This section addresses the issues raised in the fourth research question by understanding the impact of 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 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 as follows.

Max. third-party service charge per unit weight (w_6)
$<=$ (service workers travel cost+demurrage and stay costs+service cost)/(average demand across nodes)

Figure 7 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.

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

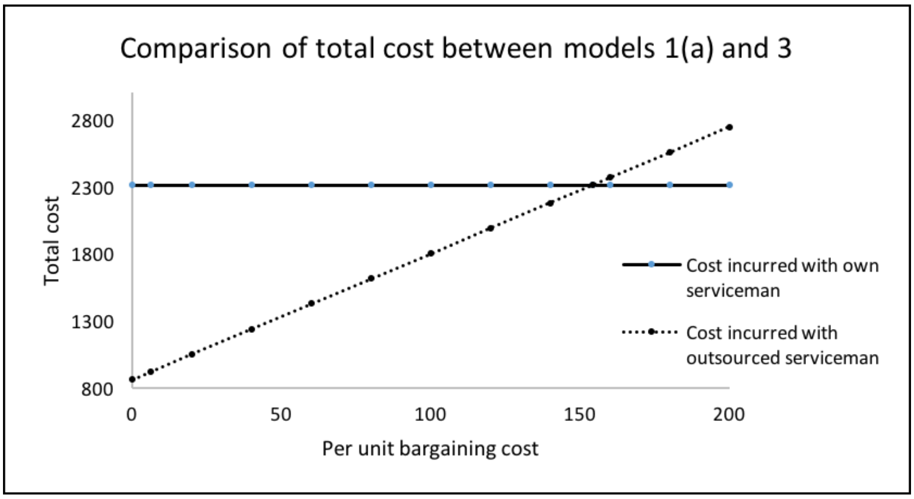


Fig. 7: Change in total cost with third-party service charge

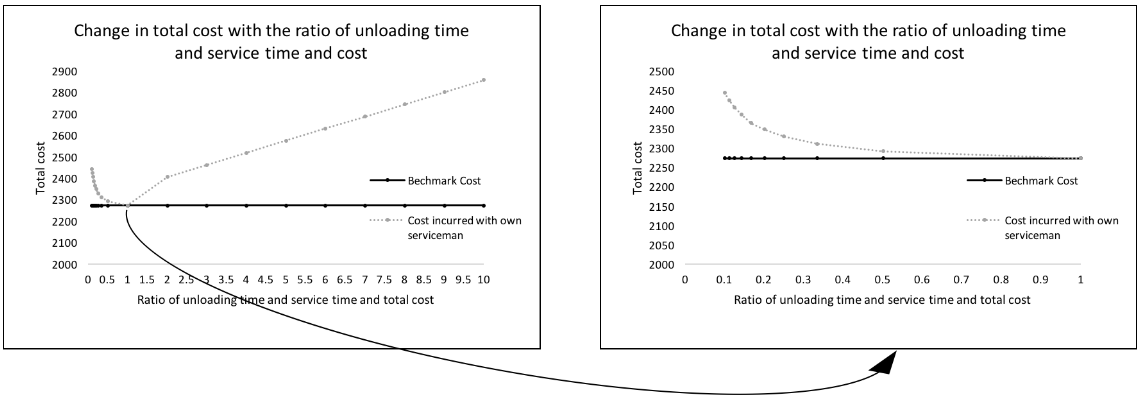


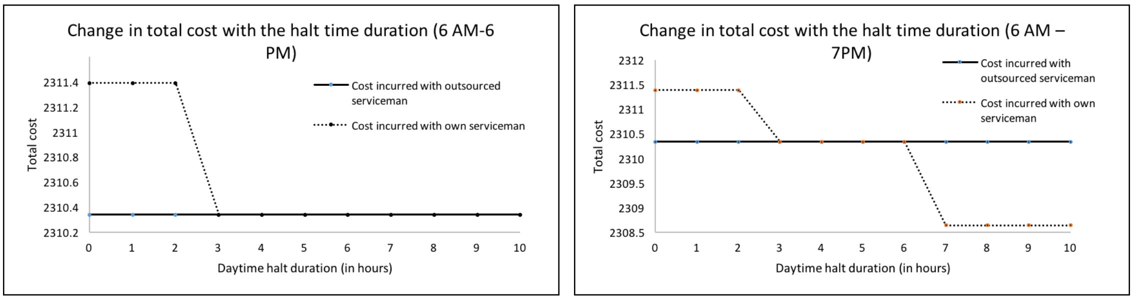
Fig. 8: (Unloading time by servicing time) vs. (total cost)

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 service worker 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 service worker must wait for the goods vehicle to arrive. Figure 8 indicates the change in total cost with the ratio value. The benchmark cost line in Figure 8 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 service person at customer locations. Figure 9 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 Figure 9(a), the end time is restricted until 6 PM, whereas in Figure 9(b), 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 service worker to take different routes. Model 2(a) followed by Model 2 will be chosen if different routes are not allowed by goods vehicle and serviceman for some firm-specific constraint.

Figure 10 summarizes this discussion by providing a flow chart 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



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

Fig. 9: Change in optimal cost with allowable halt duration and time

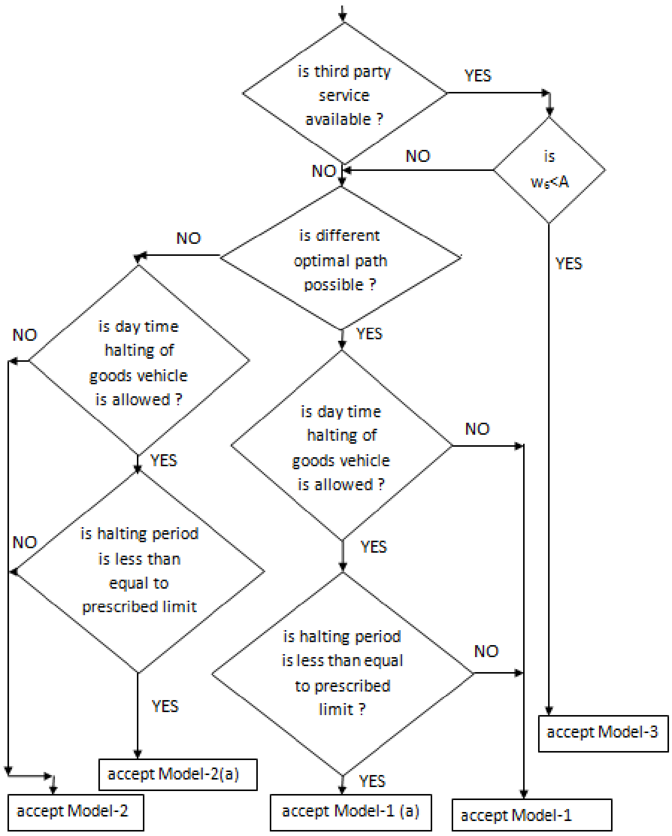


Fig. 10: Management decision flowchart

defined managerial decision-making framework that managers can use to make rational decisions with the objective of minimizing total costs.

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 is developed and implemented successfully for STSPwGDS problems. Addressing the first research question, we develop a generic formulation that can be used in various associated problem contexts. When answering the second research question on 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 travelled by the goods vehicle and service worker. The solution obtained by our algorithm also illustrates the utility of multiple vehicle types, with which the service worker chooses a vehicle type even with a higher per unit cost to reduce the penalty cost paid to the customer. To answer the third research question, the results obtained by our algorithm showcase an effective trade-off between various cost components and vehicle type. The decision-making framework, along with the developing of two model variations by incorporating halting time restrictions, effectively addresses the fourth research question. 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.

Compliance with ethical standards

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.

References

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

Averbakh, I. and Yu, W. (2018). Multi-depot traveling salesmen location problems on networks with special structure. *Annals of Operations Research*, pages 1–14.

Bailay, R. (2018). Ikea bets on small stores to go digital.

Baldwin, J. M. (1896). A new factor in evolution. *The american naturalist*, 30(354):441–451.

BBC (2018). Surrogate mother of ’twins’ finds one is hers.

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

Chang, T.-S., Wan, Y.-w., and Ooi, W. T. (2009). A stochastic dynamic traveling salesman problem with hard time windows. *European Journal of Operational Research*, 198(3):748–759.

Constantinescu, E. (2012). Three dimensional maritime transportation models.

Cordeau, J.-F., Laporte, G., Potvin, J.-Y., and Savelsbergh, M. W. (2007). Transportation on demand. *Handbooks in operations research and management science*, 14:429–466.

Corporation, X. (2012). Reliable installation service from mi.

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

Dong, G., Guo, W. W., and Tickle, K. (2012). Solving the traveling salesman problem using cooperative genetic ant systems. *Expert systems with applications*, 39(5):5006–5011.

Eiben, A. E., Raue, P.-E., and Ruttkay, Z. (1994). Genetic algorithms with multi-parent recombination. In *International Conference on Parallel Problem Solving from Nature*, pages 78–87. Springer.

Feng, Z. (2019). Constructing rural e-commerce logistics model based on ant colony algorithm and artificial intelligence method. *Soft Computing*, pages 1–10.

Focacci, F., Lodi, A., and Milano, M. (2002). A hybrid exact algorithm for the tsptw. *INFORMS Journal on Computing*, 14(4):403–417.

Forum, W. E. (2018). Where people work the longest hours.

Ghoseiri, K. and Sarhadi, H. (2008). A memetic algorithm for symmetric traveling salesman problem. *International Journal of Management Science and Engineering Management*, 3(4):275–283.

Haley, K. (1963). The multi-index problem. *Operations Research*, 11(3):368–379.

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

Lagarteja, J. G., Gerardo, B. D., and Medina, R. P. (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*, pages 3–4.

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

Ma, Y., Li, Z., Yan, F., and 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 Computing*, 23(15):6697–6714.

Maity, S., Roy, A., and Maiti, M. (2015). A modified genetic algorithm for solving uncertain constrained solid travelling salesman problems. *Computers & Industrial Engineering*, 83:273–296.

Maity, S., Roy, A., and Maiti, M. (2017). An intelligent hybrid algorithm for 4-dimensional tsp. *Journal of Industrial Information Integration*, 5:39–50.

Majumdar, J. and Bhunia, A. K. (2011). Genetic algorithm for asymmetric traveling salesman problem with imprecise travel times. *Journal of Computational and Applied Mathematics*, 235(9):3063–3078.

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

Martínez-Estudillo, A. C., Hervás-Martínez, C., Martínez-Estudillo, F. J., and García-Pedrajas, N. (2005). Hybridization of evolutionary algorithms and local search by means of a clustering method. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(3):534–545.

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

Moon, C., Kim, J., Choi, G., and Seo, Y. (2002). An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European Journal of Operational Research*, 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. and Soler, D. (2012). A new genetic algorithm for the asymmetric traveling salesman problem. *Expert Systems with Applications*, 39(10):8947–8953.

Osterman, C. and Rego, C. (2016). A k-level data structure for large-scale traveling salesman problems. *Annals of Operations Research*, 244(2):583–601.

Petersen, H. L. and Madsen, O. B. (2009). The double travelling salesman problem with multiple stacks–formulation and heuristic solution approaches. *European Journal of Operational Research*, 198(1):139–147.

Reardon, S. (2017). Genetic details of controversial 'three-parent baby' revealed. *REPRODUCTIVE BIOLOGY*, 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. *Safety science*, 103:305–315.

Roy, A., Chakraborty, G., Khan, I., Maity, S., and Maiti, M. (2016). A hybrid heuristic for restricted 4-dimensional tsp (r-4dtsp). In *International Conference on Frontiers in Optimization: Theory and Applications*, pages 285–302. Springer.

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

Skinner, M. K. (2015). Environmental epigenetics and a unified theory of the molecular aspects of evolution: a neo-lamarckian concept that facilitates neo-darwinian evolution. *Genome biology and evolution*, 7(5):1296–1302.

Tüű-Szabó, B., Földesi, P., and Kóczy, L. T. (2017). An efficient new memetic method for the traveling salesman problem with time windows. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 426–436. Springer.

Wang, J., Ersoy, O. K., He, M., and Wang, F. (2016). Multi-offspring genetic algorithm and its application to the traveling salesman problem. *Applied Soft Computing*, 43:415–423.

Wang, Y. (2015). An approximate method to compute a sparse graph for traveling salesman problem. *Expert Systems with Applications*, 42(12):5150–5162.

Wang, Y.-t., Li, J.-q., Pan, Q.-k., Sun, J., and Ren, L.-q. (2010). Memetic algorithm based on improved inver-over operator for tsp. In *Natural Computation (ICNC), 2010 Sixth International Conference on*, volume 5, pages 2386–2389. IEEE.

Wang, Z. and Lin, W.-H. (2017). Incorporating travel time uncertainty into the design of service regions for

delivery/pickup problems with time windows. *Expert Systems with Applications*, 72:207–220.
Xu, X., Yuan, H., Matthew, P., Ray, J., Bagdasar, O., and Trovati, M. (2019). Gorts: genetic algorithm based on one-by-one revision of two sides for dynamic travelling salesman problems. *Soft Computing*, pages 1–14.
Ye, T., Wang, T., Lu, Z., and Hao, J.-K. (2014). A multi-parent memetic algorithm for the linear ordering problem. *arXiv preprint arXiv:1405.4507*.

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 serviceman vehicle and only one type of goods vehicle are considered. Also values of the parameters for STSPwGDS are presented in Tables A1 to A13.

Table A1: 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	∞

Table A2: 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 A3: 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 A4: Input Data: Traveling cost per unit distance of serviceman vehicle

Crisp Travel Cost per unit distance Matrix(10×10) With Three Conveyances										
i/j	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 A5: Input Data: Traveling time of serviceman vehicle

Crisp Traveling Time Matrix(10×10) With Three Conveyances										
i/j	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 A6: Input Data: Demand/requirement of every node

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

Table A7: Input Data: Unload time and cost of every node

Crisp Unload time and cost Matrix										
i/j	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 A8: Input Data: servicing time and cost of every node

Crisp servicing time and cost Matrix										
i/j	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 A9: 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 A10: Demand/requirement of every node for M/S Sharma Furniture company

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

Table A11: Unload time and cost of every node for M/S Sharma Furniture company

Crisp Unload time and cost Matrix										
i/j	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 A12: Servicing time and cost of every node for M/S Sharma Furniture company

Crisp servicing time and cost Matrix										
i/j	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 A13: Parameter table of STSPwGDS

Parameter	Values
total demand ($\sum_{i=1}^{N} d_i = D$)	94 units
transportation of goods cost ($\sum_{i=0}^{N-1} (D - \sum d_i)\xi$)	(remaining total demand)*(1/10)
goods unload time	(demand of i^{th} node)*(1/10)
goods unload cost ($\omega(d_i) = d_i * w_4$)	(demand of i^{th} node)*(2/10)
goods vehicle holding cos ($\Pi(i) = \tau_i * \eta_2$)	(time gap in hour)*4
day time halt limit (Ω_2)	5 hours
day time halt for goods vehicle ($\Pi(i) = \tau_i * \eta_4 + \eta_5$)	(time gap in hour)*(1)+η_5
some fixed charge for day time halting (η_5)	(demand of i^{th} node)*(1/20)
serviceman stay cost ($\Pi(i) = \tau_i * \eta_1$)	(time gap in hour)*6
servicing time	(demand of i^{th} node)*(2/10)
servicing cost ($\gamma(d_i) = d_i * w_3$)	(demand of i^{th} node)*(2/10)
servicing cost by third-party ($\sigma_{cost} = d_i * w_6$)	(demand of i^{th} node)*w_6, w_6 is bargaining parameter
(consider) servicing cost per unit by third-party (w_6)	(demand of i^{th} node)*(60/10)
(consider) servicing time by third-party	(demand of i^{th} node)*(2/10)