



Multipath traveling purchaser problem with time-dependent market structure using quantum-inspired variable length genetic algorithm

Somnath Maji ^{a,*}, Kunal Pradhan ^b, Samir Maity ^c, Izabela Ewa Nielsen ^c, Debasis Giri ^d, Manoranjan Maiti ^e

^a Department of Computer Science and Engineering, Maulana Abul Kalam Azad University of Technology, NH-12, Haringhata, Nadia, West Bengal, 741249, India

^b Department of Computer Science & Engineering, Tezpur University, 784028, India

^c Operations Research Group, Department of Materials and Production, Aalborg University, Aalborg, 9220, Denmark

^d Department of Information Technology, Maulana Abul Kalam Azad University of Technology, NH-12, Haringhata, Nadia, West Bengal, 741249, India

^e Department of Applied Mathematics, Vidyasagar University, Medinipur, 721102, India



ARTICLE INFO

Keywords:

TPP
Time-dependent market structure
IoT
Type-2 fuzzy
Quantum-inspired genetic algorithm (QGA)
Multi parent crossover

ABSTRACT

In developing countries, rural and semi-urban markets remain open for a particular period of time in a day, and items' availabilities and prices vary within the opening hours of a market. Nowadays, the Internet of Things (IoT) and Type-2 fuzzy logic (T2FL) systems are used to make real-time decisions. With the logistical and infrastructural development around the world, there are several alternate routes for travel between different markets and cities. Taking the above realities into consideration, here, we develop and analyze IoT-enabled and T2FL system-based multipath traveling purchaser problems with time-dependent market structure (IoT-T2FL-MPTPPswTDMS). In this study, information regarding weather, road surface, and congestion concerning different paths obtained by the IoT are used as inputs in T2FL, and path-wise, the vehicle's average velocity is obtained using the fuzzy rules. Item availability and price in a particular market change with time. The optimization problem is, for different journey starting times, to select the appropriate markets for the purchase and the corresponding optimum routing path for minimum cost (time) against time (cost) constraint. A multi parent, varied-offspring, variable length quantum-inspired genetic algorithm (MPVOVLQiGA) with quantum selection and mutation and multi parent, varied-offspring crossover is developed to solve the proposed problems. For children, mimicking the real-life parental system, In vitro fertilization (IVF) and adopted child (if IVF failed) are considered alongwith the usual comparison crossover. The algorithm's superiority is established through a statistical test. A real-life model is numerically illustrated. The dependence on the minimum cost and time on the journey's commencement time is demonstrated.

1. Introduction

In a travelling purchaser problem (TPP), a purchaser with a vehicle starts from a depot, travels to the appropriate markets for the purchase, follows the optimum routing plan, and returns back to the depot when the demand is fulfilled. It is an *NP-hard* problem. In India, normally businessmen and individual purchasers (on specific occasions) rely heavily on the nearby rural and semi-urban markets for their purchase (cf. Obulesu, Kumar, & Babu, 2022). These markets are fundamentally fragmented with respect to (wrt) their functioning. For instance, different markets open in different time slots within a day depending on their geographical locations, customers' lifestyles and demands, etc. (as in Fig. 1). In a market, items' availabilities and prices vary within the market's operating period.

But, in the existing TPP investigations, it is assumed that markets remain open for 24 h a day, and the availabilities and prices remain constant throughout the market operating periods. But, in reality, it is not so. The markets operate in some time slots on particular days in a week decided by the local residents and traders. So, in a TPP, the purchase should be made within that period. Moreover, the 'items' availabilities and prices vary with time within the market operation period. This layered structure of a real-life TPP is demonstrated in Fig. 1, which has not been considered by earlier TPP investigators.

So far, in the routing of a TPP, it is assumed that only one route for travel between the markets is available, and based on that, the appropriate routing plan is obtained. But, with the infrastructural

* Corresponding author.

E-mail addresses: somnathmajivucs@gmail.com (S. Maji), kunalpradhan86@gmail.com (K. Pradhan), maitysamir13@gmail.com (S. Maity), izabela@mp.aau.dk (I.E. Nielsen), debasis_giri@hotmail.com (D. Giri), mmaiti2005@yahoo.co.in (M. Maiti).

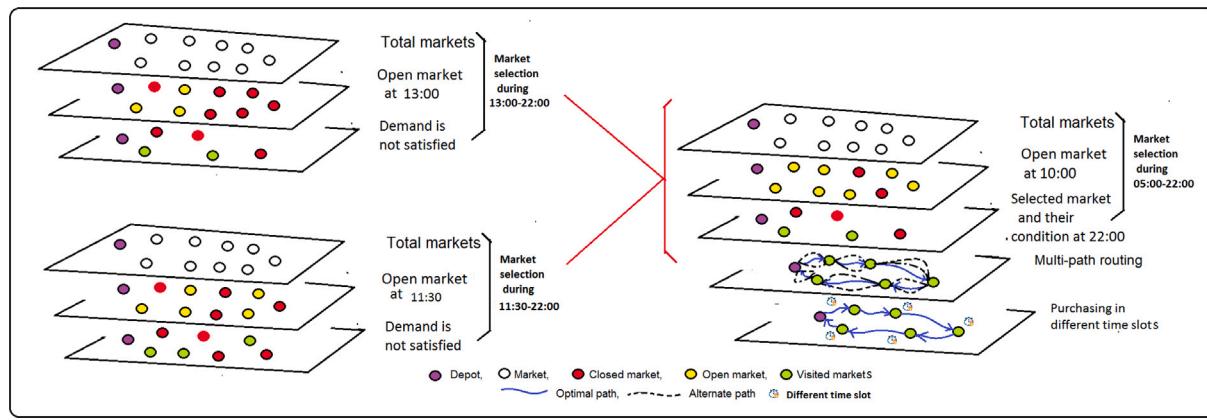


Fig. 1. Layered structure of TPP.

development throughout the world, there are several connecting roads between the markets and different cities. Among these roads, some, especially the National High Ways, which are maintained regularly from the collection of road taxes (fixed charge, say) through toll plazas, are good and smooth. Some roads are congested during certain hours or due to their geographical location. On the other hand, some roads are poorly maintained and full of ditches with uneven surfaces. Road conditions also change with the weather. Obviously, the vehicle's velocity changes with the road used for travel. Till now, none has considered the multi-path connection between the markets and the road conditions dependent travel velocity. At this point, the problem arises how to get road-specific data (congestion, etc.) throughout a long route? To the best of our knowledge, nowadays, IoTs, RFIDs are used to get these data. Then, next problem is how to get the average travel velocity (approximate) for the route? We can use the machine learning (ML), fuzzy logic (FL) or any other suitable method for this purpose. Hence a combination of IoT-ML, IoT-FL (Ali et al., 2018, Tayyaba, Ashraf, Alquthami, Ahmad, & Manzoor, 2020, Ullah, Youn, & Han, 2021) or suitable similar system is to be used to derive road-specific travel velocity. Thus, in a route, to get the road conditions at different locations, several IoTs are installed at different points of the road, which give the road conditions dynamically. From these data, the average value of road conditions is calculated.

The measured values of the road conditions through IoT are deterministic in nature. But, for the use of fuzzy logic system, in the logical relation of road conditions and travel velocity, the road characteristics are represented imprecisely (through verbal words-like good, smooth, rough, etc.), and travel velocity is defined imprecisely depending on these parameters. It is expressed by 'IF-THEN' relations (i.e., IF the road is very smooth, THEN velocity is high, say).

As mentioned earlier, until now, researchers did not evaluate vehicle's velocity by taking road surface conditions, weather, etc. into account. Here IoT and Type-2 fuzzy logic have been used for this purpose.

For the solution of the NP-hard problems, up to date, several heuristic methods are developed to have better results in a limited execution time. Genetic Algorithm (GA) was used first in 1975 and till now, several modifications to it are being made. Quantum computation has several advantages, one of which is less computational time. Very few have attempted to use Quantum-inspired GA for the routing problem.

In reality, for humans, nowadays, there are several medical arrangements to have children. Issueless people also adopt children. In GA, none has considered this idea in creating children in TPP.

In this process, some questions automatically emerge.

Research questions:

Q1: How to choose the markets which are regular but operate only during a time slot of a day?

Q2: If the price of an item goes down within the operating period of the market, how to take advantage of the reduced price?

Q3: If the items' availabilities also vary within the markets' period, how to utilize the maximum availability situations and make the trade-off between the availability and price?

Q4: Journey's starting time is important in the time-dependent-market structure (TDMS), because the accessibility to the market changes with time. Then, what is the best journey starting time in TPP for a TDMS?

Q5: In this context, what will be the routing plan of a purchaser if the real-life road conditions (weather, congestion, and surface) are considered and monitored by the IoT?

Q6: Now, the question arises: how to derive the appropriate travel velocity out of these crisp data from IoT?

Some conventional TPP problems are solved using exact approaches (see Table 1). However, large-scale problems take longer computational time, so we need robust solutions and techniques in a reasonable time. As TPP is an NP-hard problem, few researchers solved the TPPs using quantum inspired GA (QiGA) (Pradhan, Basu, Thakur, Maity, & Maiti, 2020), noble GA (Roy, Gao, Jia, Maity, & Kar, 2020), etc. In this context, some research questions arise.

Q7: How to solve the TPP with TDMS using heuristic methods—say, Genetic Algorithm (GA)? As demand is fixed and availabilities at markets are different, the lengths of chromosomes in GA will be obviously variable.

Q8: How to formulate a variable length GA with multiple parents and varied off-springs to solve IoT-T2FL-MPTPPswTDMS? Will it be faster if the quantum concept is introduced in the GA?

To answer the Questions 1–4, we formulate a multi-path TPP with time-dependent market structure (MPTPPwTDMS). For Questions 5 and 6, there are two options-(i) Use of IoT to have real-life road conditions and get the appropriate travel velocity using the build-in AI software in IoT (normally machine learning) or (ii) Use the IoT to collect the real-life road-specific data and using those, derive the corresponding travel velocity using a fuzzy logic controller (maybe Type-2 fuzzy logic one). In the present study, we have used IoT and Type-2 fuzzy logic controller together for the above purpose.

Thus an IoT-enabled Type-2 fuzzy logic system-based multipath TPP with time-dependent market structure (IoT-T2FL-MPTPPwTDMS) is formulated. Till now, none considered the TPP with TDMS. Questions 7 and 8 prompted us to formulate a multi parent, varied-offspring variable length quantum-inspired genetic algorithm (MPVOVLQiGA) with quantum selection and mutation and multi parent varied-offspring crossover to solve the proposed IoT-T2FL-MPTPPswTDMS.

Here, we formulate and solve the IoT-enabled and Type-2 fuzzy logic system-based multipath traveling purchaser problems with time-dependent market structure (IoT-T2FL-MPTPPwTDMS) for minimum

Table 1
Literature survey of this investigation.

References	Models	Applied methods	Objective & Constraint			Parameters
			Exact Heuristics	Meta-heuristics	Q-meta-heuristics	
Variant's of TPP						
Choi and Lee (2011)	Multiple TPP	✓			✓	
Manerba and Mansini (2012)	Capacitated supplier selection		✓		✓	
Batista-Galván, Riera-Ledesma, and Salazar-González (2013)	Double TPP	✓			✓	
Bianchessi, Mansini, and Speranza (2014))	CTPP with multiple vehicle	✓			✓	
Manerba and Mansini (2014)	CTPP with quantity discount	✓ ✓			✓	✓
Manerba and Mansini (2015)	CTPP with incompatibility	✓			✓	✓
Manerba, Mansini, and Riera-Ledesma (2017)	CTPP and UTTP variants	✓ ✓ ✓	✓		✓ ✓ ✓ ✓	✓
Angelelli, Gendreau, Mansini, and Vindigni (2017)	Time-dependent CTPP	✓ ✓			✓ ✓	
Bernardino and Paías (2018))	TPP with decision hierarchy	✓ ✓			✓	
Palomo-Martínez and Salazar-Aguilar (2019)	Bi-objective TPP	✓			✓ ✓	
Pradhan et al. (2020)	Solid TPP	✓ ✓ ✓			✓ ✓	
Roy et al. (2020)	Solid TPP	✓ ✓ ✓			✓ ✓	✓
Cuellar-Usaquén, Gómez, and Álvarez-Martínez (2021)	TPP	✓ ✓			✓	
Bianchessi, Irnich, and Tilk (2021)	CTPP with multiple vehicle	✓ ✓			✓ ✓	
Kucukoglu (2022)	CTPP with fast service	✓ ✓			✓ ✓ ✓	
Khodadadian, Divsalar, Verbeeck, Gunawan, and Vansteenwegen (2022)	TPP with time dependent	✓			✓ ✓	
Present investigation (2023)	Multipath TPP with time dependent	✓ ✓ ✓			✓ ✓	✓ ✓

cost (time) with a time (cost) constraint. Here it is assumed that different rural and semi-urban markets remain open for different periods (fixed) of time in a day. Availability and price of an item change during the market's period. Due to infrastructural developments, in each country, there are several (more than one) connecting paths for travel among the markets and between the markets and cities. Some of these paths are very smooth and good, some others very bad, congested, damaged due to excessive rain, etc. Nowadays, IoT and Type-2 fuzzy logic systems are used to predict the real-time velocity of a travel vehicle. The optimum tour plan for visiting a subset of markets to satisfy the discrete demand of a product is found so that the system cost or time is minimum. For solution, a multi parent, varied-offspring, variable length quantum-inspired genetic algorithm (MPVOVLQiGA) with quantum selection and mutation and multi parent, varied-offspring crossover is developed. The superiority of the intended algorithm is established by testing the benchmark TPP instances using a statistical test. Some managerial considerations are discussed.

Thus, the contributions of this investigation are as follows:

- Proposed an IoT-enabled and Type-2 fuzzy logic system-based multipath traveling purchaser problem with time-dependent market structure (IoT-T2FL-MPTPPswTDMS) is having traveling, purchasing, fixed charge, loading, and parking costs. Here, markets remain open at different time slots in a day. Depending on road characteristics, IoT-Type-2 fuzzy logic-based velocity prediction is incorporated.
- A multi parent, varied-offspring, variable length quantum-inspired genetic algorithm (MPVOVLQiGA) is developed and used to solve IoT-T2FL-MPTPPswTDMS.
- Quantum selection and mutation and multi parent varied-offspring crossover, Adoption of offsprings is also considered.
- A framework for policy-level decision-making is developed through scenario analysis. It is illustrated that minimum cost or time depends on the journey starting time.

The paper is organized as follows: Sections 1 and 2 present a brief introduction and literature review respectively. The mathematical formulation of the proposed model is presented in Section 3. Section 4 explains the intended algorithm (MPVOVLQiGA). The computational experiment is done through Section 5. The dissection of the results is performed in Section 6. In Section 7, the conclusion along with limitations and future scope is presented.

2. Literature review

A concise literature development of TPP wrt different problem formulation and solution characteristics during the last decade is presented chronologically in Table 1.

2.1. Model related

First, Ramesh (1981) presented TPP as an extension of the traveling salesman problem (TSP). After that, several researchers investigated different forms of TPP, presented in Table 1. Khodadadian et al. (2022) introduced time windows and service time-based profit maximized TPP as a mixed integer programming model. They assigned minimum and maximum service times at each market under some time constraints. They solved it through a meta-heuristic algorithm, which is based on the variable neighborhood search (VNS) method. Parking charges play a significant role in routing (Albalate & Gragera, 2020, Yang, Chen, Ye, Chen, & Luo, 2023). Till now, problems arised due to parking is not considered in the formulation of TPPs. In this investigation, market location and waiting time due to parking are taken into account while determining the parking charges. Angelelli et al. (2017) investigated TPP with mixed integer formulation based on time-varying decreasing (constant rate) availability of quantities of purchased items. To solve this problem and standard TSPLIB instances, they used a branch-and-cut framework (with branching strategy and primal heuristic enforcement). Also, they used CPLEX 12.6 to solve this problem. So far, none considered a time-dependent market structure in TPP, where availabilities and prices in different markets depend on the market's age. So, in this investigation, we consider this a realistic approach. Pradhan et al. (2020) introduced a solid TPP, i.e., they consider multi-vehicle between arbitrary two markets. But in this investigation, we incorporate multipath among the markets and depot and also consider quantum path selection between arbitrary two markets among the available multipaths. The vehicle's route velocity plays a significant role in reaching a market. The vehicle's velocity is predicted using machine learning (Liu, Asher, Gong, Huang, & Kolmanovsky, 2019), artificial neural networks (Gaikwad, 2020), etc. The information collections and decision making through IoTs for waste collection problem proposed by Roy, Manna, Kim, and Moon (2022). Till now, none took IoT-Type-2 fuzzy logic combination into account while predicting a vehicle's speed. We predict the vehicle's speed using IoT-Type-2 fuzzy logic for a more accurate prediction. Choi and Lee (2011) introduced fixed cost with respect to vehicle operation.

2.2. Solution methods related

To solve different versions of TPP, different exact and meta-heuristic algorithms are used, which are presented in Table 1. To solve the TPPs through exact methods, Cuellar-Usaquén, Gomez, and Álvarez-Martínez (2023) proposed an adaptive large neighborhood search, Manerba et al. (2017) applied branch-and-bound method, the branch-and-cut methods were used by Bianchessi et al. (2021), and dynamic programming approach by Cuellar-Usaquén et al. (2023). As TPP is an NP-hard problem (Mor, Shabtay, & Yedidsion, 2021, El Khadiri, Yeh, & Cancela, 2023, Kang, Kung, Chiang, & Yu, 2023), meta-heuristic methods (Wang & Alidaee, 2023, Tabrizi, Vahdani, Etebari, & Amiri, 2023, Jiang, Lu, Ren, Cheng, & Liu, 2023) are suitable for large and complex TPPs wrt computation time. Focusing on that, several researchers used GAs to solve TPPs. Manerba et al. (2017) presented all those papers through a survey. Roy et al. (2020) introduced a noble GA with probabilistic selection, In Vitro Fertilization (IVF) crossover, and random mutation for solving solid green TPP. Pradhan et al. (2020) used a quantum-inspired genetic algorithm with IVF crossover and mutation operator. They considered carbon credit/penalty structure and used multiple vehicles for effective results. For solving TPP, GA should be developed with variable length chromosomes as demand may be satisfied with different numbers of markets. Focusing on that, we introduce variable length GA (VLGA) for the proposed IoT-T2FL-MPTPPswTDMS models. Variable length chromosome in GA (VLGA) was used to solve some routing-specific problems (Cruz-Piris, Marsa-Maestre, & Lopez-Carmona, 2019, Ha, Dao, Pham, & Le, 2021, Xiao, Yan, Basodi, Ji, & Pan, 2020, Keshavarzi, Haghigat, & Bohlouli, 2021, Park, Shin, Shin, Chi, et al., 2019, etc.). But, to the best of our knowledge, none considered VLGA in TPP. As in each routing, demand may be satisfied with a different number of markets, so VLGA plays a crucial role in solving TPP. Considering the computational time, Quantum inspired heuristic algorithms are more effective (Papalitsas, Kastampolidou, & Andronikos, 2021), and focusing on that, researchers developed quantum GA (Wang & Wei, 2021, Cheng, Wang, & Xu, 2020, Zhu, Xiong, & Liang, 2018, Li, Hu, Zhu, Lei, & Xing, 2019, Mousavi, Afghah, Ashdown, & Turck, 2019, Wan, He, Wang, Yan, & Shen, 2019, Pradhan et al., 2020, etc.). IVF/multi parent crossover in GA is more suitable than the two-parent crossover as more information is used to generate offspring. Some of these investigations (Roy et al., 2020, Pradhan et al., 2020, Das, Roy, Maity, Kar, & Sengupta, 2021, Arram & Ayob, 2019, etc.) are available in the literature. Varied-offspring crossover is also helpful in maintaining the diversity of the solutions. Some of those investigations are Katoch, Chauhan, and Kumar (2021), Wang, Zhang, Ersoy, Sun, and Bi (2019), etc. They did not consider varied-offspring.

In this investigation, varied-offspring are considered with the adoptive child concept. Till now, none considered this approach. Also, we design problem-specific crossover and Quantum mutation, as the proposed model is based on a time-dependent market structure, and the markets may remain open in different time slots. It is to be noted that these crossover and mutation operations are performed within the feasible time (market operating time). So, to solve the NP-hard IoT-T2FL-MPTPPswTDMS, we use a multi parent, varied-offspring, variable length quantum-inspired genetic algorithm (MPVOVLQiGA) with quantum selection, multi parent varied-offspring crossover, and quantum mutation.

3. Formulation of IoT-T2FL-MPTPPswTDMS

3.1. Nomenclature

Notations and corresponding descriptions are presented in Table 2.

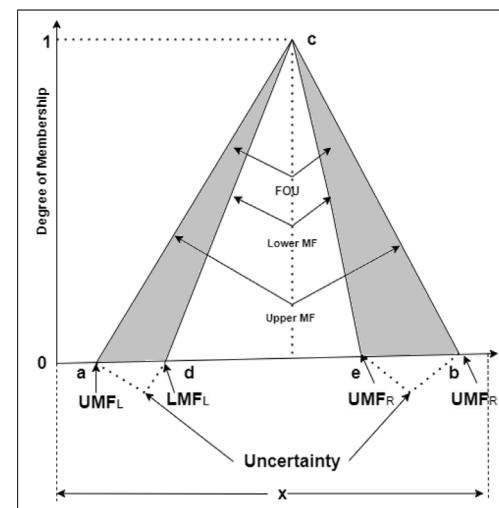


Fig. 2. Type-2 fuzzy membership set.

3.2. Assumptions/constraints

- (i) Total availability is greater than or equal to total demand.
- (ii) Purchaser visits a market only once and purchases the whole amount available in the market at the purchaser's visiting time. In the last visiting market, only the required amount to meet the demand is purchased.
- (iii) At a time (not fixed), the purchaser starts from a depot with a vehicle of sufficient capacity and comes back to the same depot after purchase.
- (iv) Purchaser does not have any prior idea/information about the availabilities and prices at different markets.
- (v) Different markets are operated at different time slots in a day and there are several markets available for purchase around the depot.
- (vi) Availabilities and prices of items in a market vary during the market period i.e. at the beginning of the market, availability (price) of an item is medium (high), becomes high (medium) after some time, and then low (low) towards the end.
- (vii) There are several connecting paths (three in this investigation) between the markets and among the markets and depot.

3.3. Mathematical preliminaries

The concept of type-2 fuzzy and type-2 fuzzy logic are given below.

3.3.1. Interval type-2 fuzzy set (IT2FS) (c.f Castillo, Melin, Kacprzyk, & Pedrycz, 2007; Ullah et al., 2021)

A type-2 fuzzy set (T2FS) expresses the nondeterministic truth degree with imprecision and uncertainty for an element that belongs to a set. A T2FS denoted by \tilde{A} , is characterized by a type-2 membership function $\mu_{\tilde{A}}(x, u)$, where $x \in X$, $u \in J_x^u \subseteq [0, 1]$ and $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$ defined in Eq. (1). Mendel (2017)

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\} = \{(x, \mu, \mu_{\tilde{A}}(x, u)) | \forall x \in X, \forall u \in J_x^u \subseteq [0, 1]\} \quad (1)$$

This is also expressed as

$$\tilde{A} = \int_{x \in X} \int_{\mu \in J_x^u} \mu_{\tilde{A}}(x, \mu) / (x, \mu) J_x \subseteq [0, 1] \quad (2)$$

where $0 \leq \mu_{\tilde{A}}(x, u) \leq 1$, and $\int \int$ are the combination of given x and μ . J_x indicates primary membership of \tilde{A} where $J_x \subseteq [0, 1]$ for $x \in X$. From Eqs. (1), (2) and $J_x \subseteq [0, 1]$ gives a restriction that is equivalent to

Table 2
Notation and description of parameters and decision variables.

Notation	Description
R	Set of available alternate path
dis_{ijr}	Distance from i th market to j th market using $r \in R$ path
c_{ijr}	Traveling cost per unit distance from i th market to j th market using $r \in R$ path
ϕ_{ijr}	Fixed cost from i th market to j th market using $r \in R$ path
$P_{vel_{ijr}}$	Predicted velocity from i th market to j th market using $r \in R$ path
σ_{ijr}^{ys}	Real-time value of g th road parameters from y th IoT device via i th to j th market using $r \in R$ path
q_{ik}	Availability of the k th product at the i th market
p_{ik}	Purchase cost of the k th product at the i th market
z_{ik}	Purchase quantity of the k th product at the i th market
λ_{ik}	Loading cost per unit weight of the k th product at the i th market
α_{ik}	Purchasing time per unit weight of the k th product at the i th market
β_{ik}	Loading time per unit weight of the k th product at the i th market
γ_{ik}	Parking cost per unit weight per unit time of the k th product at the i th market
T_i^Q	Cumulative time taken up to the i th market
R_{ij}	Available paths between i th and j th markets
x_{ijr}	Binary decision variable, $x_{ijr} = 1$ for the travel from i th market to j th market, else, $x_{ijr} = 0$
y_i	Binary decision variable, $y_i = 1$ for i th market visited, else, $y_i = 0$
$-i(a) \frac{P_{vel}}{a} j(b) -$	Travel from the i th and j th market with predicted velocity (P_{vel}) through path a , $a = 0, 1, 2$
g	Road parameter (weather, congestion, road surface)
t_j^{sp}	Total spent time of the purchaser at the j th market
σ_{ijr}^{ys}	Total spent time of the purchaser at the j th market
D_k	Total purchase quantity of k th product
$t_{s,s+1}$	Travel time from s th to $(s+1)^{th}$ market
u_{ir}	Departing from i th market through r th route
t_j^{in}	In time of the purchaser from j th market
t_j^{ot}	Out time of the purchaser from j th market
j^o	Opening time of the j th market
j^c	Closing time of the j th market
t_j^p	Purchasing time at the j th market
t_j^l	Loading time at the j th market
p_{jk}^1	Opening price of the k th product at the j th market
p_{jk}^3	Closing price of the k th product at the j th market
q_{jk}^1	Opening availability of the k th product at the j th market
q_{jk}^3	Closing availability of the k th product at the j th market

wrt: With respect to

$0 \leq \mu_{\tilde{A}}(x, u) \leq 1$ for type-1 membership function (T1FSMF). For $x = x'$, and each value of x :

$$\mu_{\tilde{A}}(x') = \sum \mu \in J_{x'} f x'(\mu) / \mu, \text{ for } \mu \in J_{x'} \subseteq [0, 1] \quad (3)$$

and $x' \in x$

$$FOU \tilde{A}(x') = \bigcup_{\forall x \in X} J_{x'} = \{(x, \mu) : \mu \in J_x \subseteq [0, 1]\} \quad (4)$$

According to the above Eqs. (3) and (4), $J_{x'}$ and $\mu_{\tilde{A}}(x')$ stated a primary and secondary membership functions of x and the together of all primary MFs denoted through the footprint of uncertainty (FOU). The T2FS contains two T1FS MF with bounds on $FOU(\tilde{A})$, a lower bound $\underline{\mu}_{\tilde{A}}(x)$, the upper bound $\overline{\mu}_{\tilde{A}}(x) \forall x \in X$. Now these are given below

$$\underline{\mu}_{\tilde{A}}(x') \equiv \overline{FOU(\tilde{A})} \mid \forall x \in X \quad (5)$$

and

$$\overline{\mu}_{\tilde{A}}(x') \equiv \underline{FOU(\tilde{A})} \mid \forall x \in X \quad (6)$$

In interval T2FS

$$J_X = [\underline{\mu}_{\tilde{A}}(x), \overline{\mu}_{\tilde{A}}(x)], \forall x \in X \quad (7)$$

In the given Fig. 2, showing that FOU (shaded region), lower MF (LMF) and upper MF (UMF) which are presented by five points

(a, b, c, d, e) and four linear functions. The centroid of T2FMF is formulated as

$$\begin{aligned} Triangular(x; a, b, c, d, e) &= \max(0, \min(T_1, T_2, e)) \\ UMF &= T_1(x; a, b, c) \\ LMF &= T_2(x; d, e, c) \end{aligned} \quad (8)$$

$$c = \frac{\sum_{i=1}^q \mu(x_i)x_i}{\sum_{i=1}^q \mu(x_i)} \quad [c_1, c_2] = \left[\frac{\sum_{i=1}^q \mu'(x_i)x_i}{\sum_{i=1}^q \mu'(x_i)}, \frac{\sum_{i=1}^q \mu''(x_i)x_i}{\sum_{i=1}^q \mu''(x_i)} \right] \quad (9)$$

where $\mu''(x_i)$ and $\mu'(x_i)$ are the given value of UMF and LMF, which maximizes and minimizes the weighted average. The type-2 fuzzy set can handle complex and higher order uncertainty.

3.3.2. Type-2 fuzzy logic

The Mamdani IT2FIS (Castro, Castillo, & Martinez, 2007), is designed with n inputs, m outputs and r rules. The k th rule with interval type-2 fuzzy antecedents $\tilde{A}_{k,j} \in \{\mu_{i,l_{k,j}}\}$, interval type-2 fuzzy consequent $\tilde{C}_{k,j} \in \{\sigma_{j,l_{k,j}}\}$ and interval type-2 fuzzy facts \tilde{A}_i are inferred as a direct reasoning. The evaluation of this type of reasoning follows the formulation given by Castillo et al. (2007).

In this investigation, road specification data obtained by the IoT are crisp input in the T2FL. The crisps data are mapped by fuzzifier and sent to the T2FL rule to apply the “IF-THEN” rule. All the values of T2FS outputs in converted to T1FS outputs by type-reducer. Defuzzification further converts the T1FS onto a crisp value, which is approximately near the average of the right-end and left-end points (c.f Fig. 4). this process is mathematically illustrated at the end of Section 3.4.

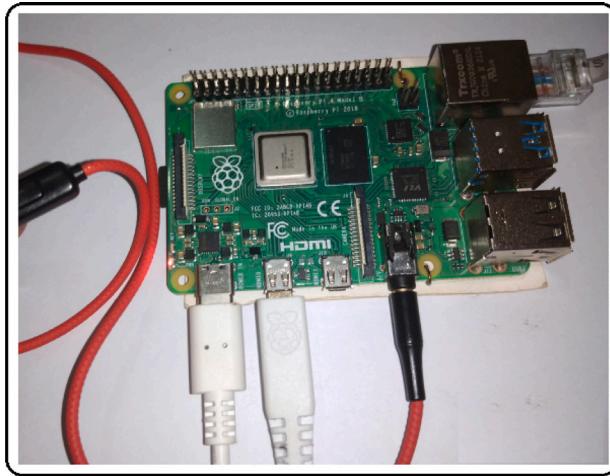


Fig. 3. IoT using Raspberry pi.

3.4. IoT enabled and type-2 fuzzy logic controller based information

IoT devices:

In this experiment, we use a Raspberry Pi. The crisp values of weather, road conditions, and congestion are collected through Raspberry Pi (Fig. 3).

The some sensors are used to fetch the data along with Raspberry pi, which is a small Linux-based computer that mainly uses Python for various tasks.

These are, Ultrasonic sensors: (Ultrasonic sensors tally vehicles, sense presence, and emit waves for occupancy detection.), Proximity sensor: (Proximity sensor detects nearby objects without direct contact through sensing presence/approach.), GPS: (GPS offers positioning, navigation, and timing services; likewise, a temperature sensor measures temperature.), Temperature sensor: (gauges heat, cold, converts to electrical signal for detection and measurement.), Proximity sensor (Proximity sensor detects nearby objects without physical contact by sensing presence.), Rain Drop Sensor (Raindrop sensor outputs digital signal and analog measurement upon moisture threshold.), BMP180: (High-precision sensor calculates pressure and temperature, accounting for gas density changes.), DHT-11: (Applied widely, it measures temperature, humidity in HVAC, aiding weather forecasts.), and Road surface condition sensors: (Collecting weather data: temperature, surface condition, dew point, road grip.)

Type-2 Fuzzy logic controller (Type-2 FLC): A Type-2 Fuzzy Logic Controller (Fig. 4) is an advanced variant of fuzzy control systems. It deals with uncertainties in a more robust manner by handling uncertain information. Unlike Type-1 Fuzzy Logic, Type-2 introduces higher-order uncertainty, allowing better representation of complex systems. This controller finds applications in areas requiring enhanced decision-making under severe uncertainties, such as complex industrial processes and autonomous systems.

In fuzzy logic system, input (premises) and output (conclusion) are connected with fuzzy rules (Implication). These premises and conclusions are expressed in fuzzy sense through verbal words (cf. Table A.1). Thus, in fuzzy controller, the sequence of operations is as follows:

Crisp input is fed first. Then it is fuzzified using normalization, if required. After this, fuzzy rules are applied and fuzzy output is obtained. For the user, the fuzzy output is defuzzified and its crisp value is obtained (cf. Fig. 4).

For the present investigation, crisp values of weather and road surface conditions including confession are collected through several IoTs placed at different locations of a route. From the collected data, the average values of the above parameters are obtained for a route,

which is fed into the FLC to get the vehicle's average velocity for travel along that route.

Data collection through IoT devices:

Different sensors are presumably installed on the roads leading to the markets, and each smart device is linked to the others via the internet. Here, routing is done by a smart vehicle. Before commencing a trip from one market (or depot) to another, it collects real-time data from IoT sensors about the weather and road conditions along the way. The average data regarding road conditions obtained through IoTs is put as input in a type-2 fuzzy controller (cf. Castro et al., 2007); and as an output, the corresponding crisp velocity is predicted, and the purchaser travels at that velocity.

Let us say there are n devices connected to the Internet of Things positioned along the route from the i th market to the j th one via the r th route. Let σ_{ijr}^{yg} be the real-time values of the g th road parameters, such as weather ($g = 1$), congestion ($g = 2$) and road surface ($g = 3$) from y th IoT devices, σ_{ijr}^y . Then the average/mean value ($m(\sigma_{ijr}^g)$) of the g th parameter through the r th route journey in between the i th and j th markets is as follows: (Eq. (10)).

$$m(\sigma_{ijr}^g) = \frac{\sum_{y=1}^n \sigma_{ijr}^{yg}}{n}, \quad g = 1, 2, 3 \quad (10)$$

Type-2 Fuzzy logic-based velocity prediction using IoT furnished data:

Fuzzy logic has been used for simulation. Outputs from IoT devices comprising sensors are taken as the inputs of the fuzzy controller. Rules are developed based on the conditions and requirements to make decisions as output.

In this model, to predict the velocity of the used vehicle, IoT and Type-2 fuzzy logic systems (Fig. 4) are used. First, through IoT (using the sensors) for different parameters, we get the input parameters of weather, road surface, and congestion for Type-2 fuzzy logic system (Fig. 5), and the crisp value of velocity obtained as output using the fuzzy logic rules given in Table A.1

Weather is categorized into five parts very good, good, medium, bad, and very bad (membership function) in the range [0,1]. Similarly, road surfaces are categorized as very smooth, smooth, medium, rough, and very rough and road congestion as very rare, rare, medium, often, and very often (membership function) in the range [0,1]. This is illustrated in Fig. 5(a). The velocity of the vehicle is evaluated using Mamdani Type-2 fuzzy logic based on weather, road surface conditions, and congestion nature (Fig. 5(a)), and using 25 rules (Table A.1). Here, the velocity of the vehicle is the output parameter within [15-65].

It is seen that the velocity becomes more dynamic (in Figs. 5(c), 5(d)) using the set of rules in the Type-2 fuzzy logic. Using the interval value viewer, for different values of Weather, Road surface, and Congestion, we get the corresponding Velocity from Fig. 5(b).

This above process is numerically illustrated in Fig. 6.

3.5. Mathematical formulation of IoT-T2FL-mptppswtdms

Following the assumptions in Section 3.2, we formulated the IoT-T2FL-MPTPPswTDMS for minimum cost (time) with time (cost) constraint, which is illustrated in Fig. 7.

Here, the purchaser starts from the depot at a particular time, visits the markets to purchase the product, and returns to the depot when the purchased amount equals the demand.

To explain in detail, let the purchaser leave the depot at 5 o'clock. He first finds the markets open at 5 o'clock-let these are A, B, and C. One of these markets, say B, is selected randomly, and the purchaser reaches market B at 7.30 o'clock, provided market B is still open until 11 o'clock. The available amount of the item at B at that moment is purchased at the prevailing price. Let purchasing and loading at market B be completed by 7.30 o'clock. Now, the purchaser again detects the available markets opened at 7.30 o'clock. The above procedure is repeated till the demand is fulfilled, and after that, the purchaser returns to the depot.

In this investigation, four IoT-T2FL-MPTPPswTDMS are formulated and solved following the classification in Fig. 8.

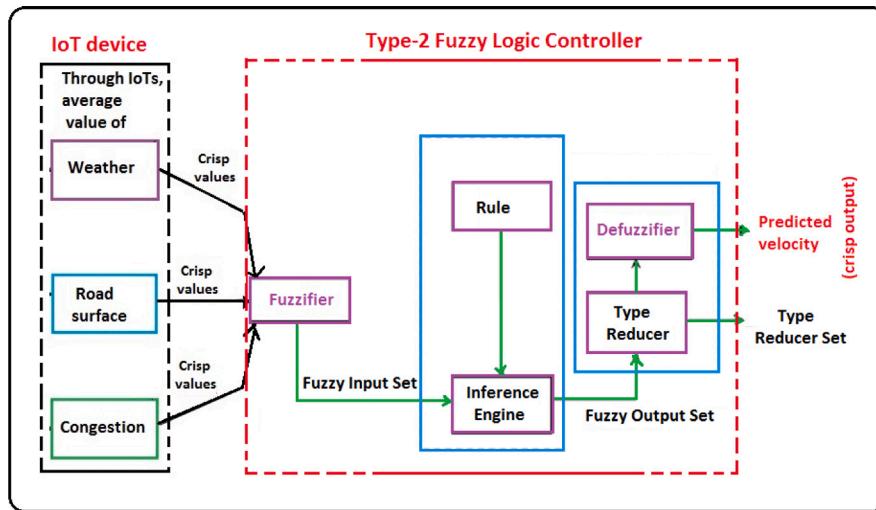
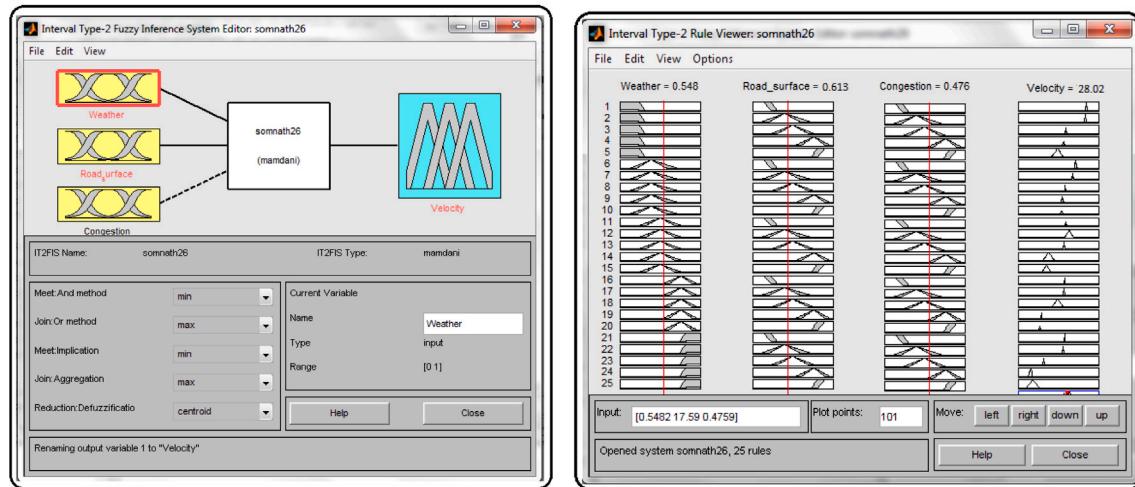
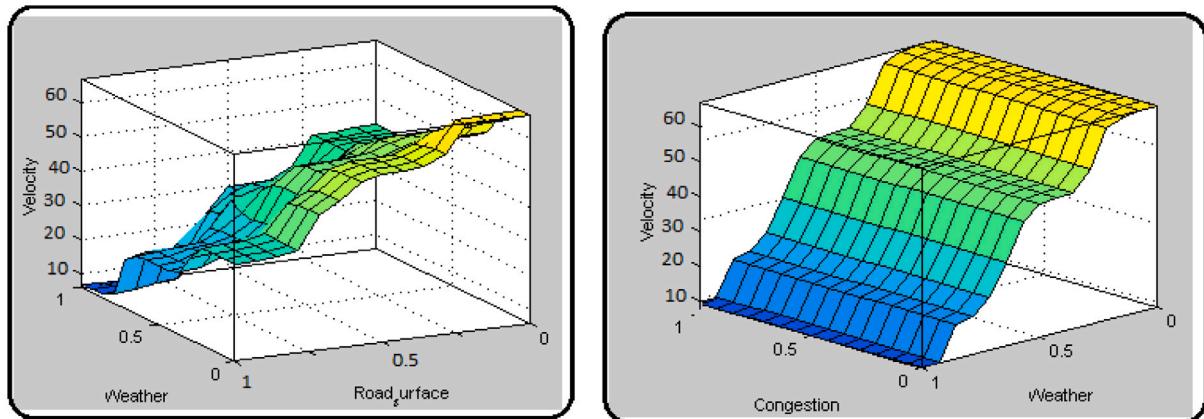


Fig. 4. Block diagram of IoT-fuzzy system.



(a) Overview of Type-2 fuzzy logic

(b) Interval rule viewer



(c) Surface view (velocity, weather, roadsurface)

(d) Surface view (velocity, congestion, weather)

Fig. 5. Graphical view of Type-2 fuzzy logic.

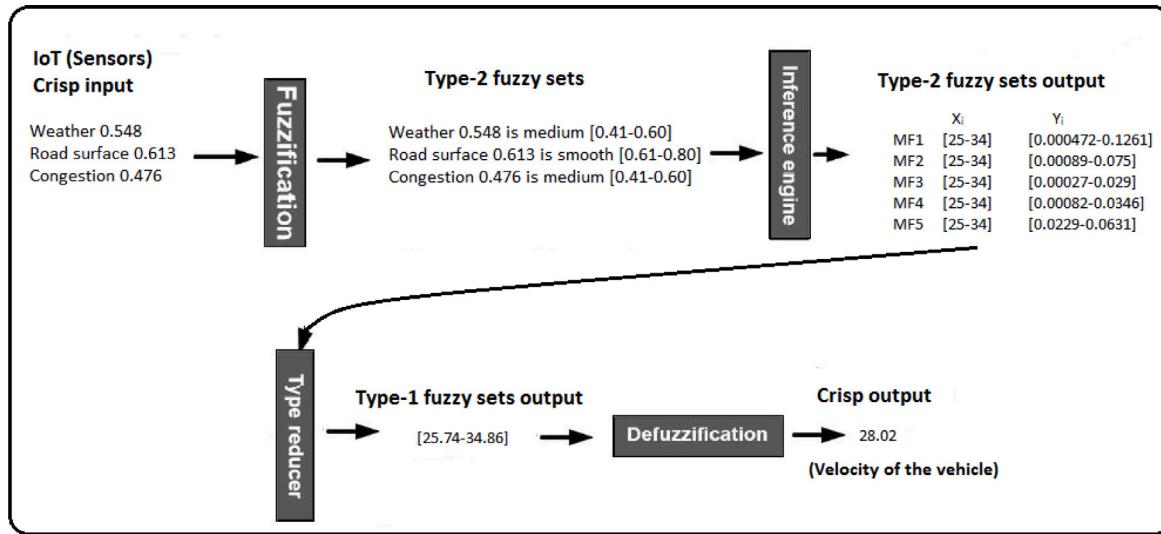


Fig. 6. Example of Type-2 fuzzy logic.

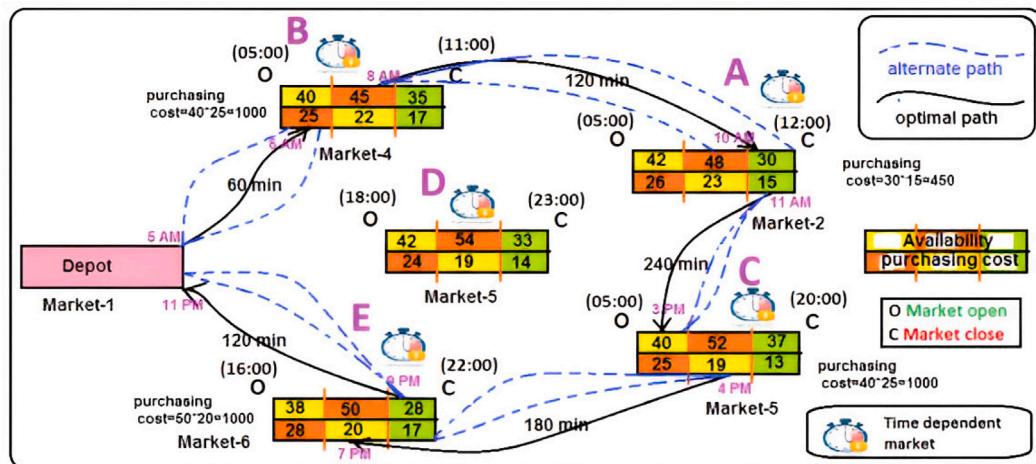


Fig. 7. Graphical representation of model.

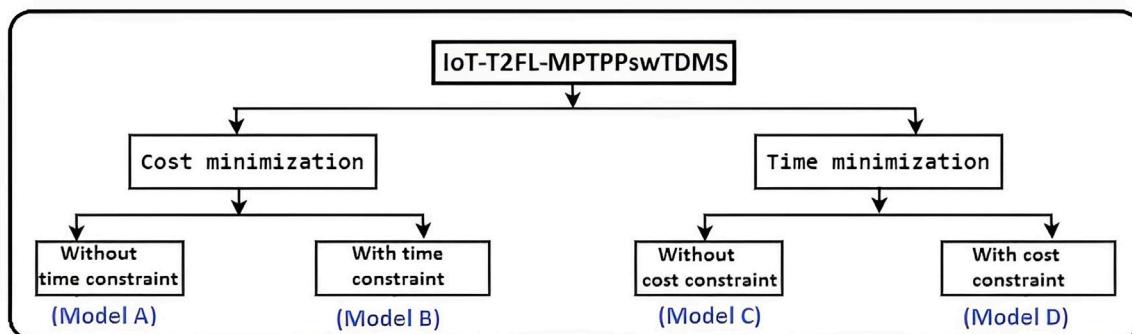


Fig. 8. All models at a glance.

3.5.1. Mathematical formulation of model A: Cost minimization IoT-T2FL-MPTPPswTDMS

$$\text{Min } Z = \underbrace{\sum_{i \in V} \sum_{j \in V} \sum_{r \in R} c_{ijr} dis_{ijr} x_{ijr}}_{\text{Traveling Cost}} + \underbrace{\sum_{i \in M} \sum_{k \in K} p_{ik} z_{ik}}_{\text{Purchasing Cost}} + \underbrace{\sum_{i \in V} \sum_{j \in V} \sum_{r \in R} \phi_{ijr} x_{ijr}}_{\text{Fixed Cost}} \\ + \underbrace{\sum_{i \in V} \sum_{k \in K} z_{ik} * \lambda_{ik}}_{\text{Loading Cost}} + \underbrace{\sum_{i \in V} \sum_{k \in K} (\alpha_{ik} + \beta_{ik}) * z_{ik} * \gamma_{ik}}_{\text{Vehicle Parking Cost}} \quad (11)$$

$$T = \underbrace{\sum_{i \in V} \sum_{j \in V} \sum_{r \in R} (dis_{ijr} / P_{vel_{ijr}}) x_{ijr}}_{\text{Traveling Time}} + \underbrace{\sum_{i \in V} \sum_{k \in K} z_{ik} * \alpha_{ik}}_{\text{Purchasing Time}} + \underbrace{\sum_{i \in V} \sum_{k \in K} z_{ik} * \beta_{ik}}_{\text{Loading Time}} \quad (12)$$

$$\text{subject to } \sum_{i \in M_k} z_{ik} = D_k, \quad k \in K \quad (13)$$

$$z_{ik} \leq q_{ik} y_i \quad k \in K, i \in M \quad (14)$$

$$\sum_{j \in V, j \neq i} \sum_{r=1}^{R_{ij}} x_{ijr} = y_i, \quad i \in M \quad (15)$$

$$\sum_{i \in V, i \neq j} \sum_{r=1}^{R_{ij}} x_{ijr} = y_j, \quad j \in M \quad (16)$$

$$\sum_{i \in V} \sum_{r=1}^{R_{i0}} x_{i0r} = 1 \quad (17)$$

$$\sum_{j \in V} \sum_{r=1}^{R_{0j}} x_{0jr} = 1 \quad (18)$$

$$x_{ijr} \in \{0, 1\}, \quad i, j \in V, \quad r \in R_{ij}, \quad i \neq j \quad (19)$$

$$y_i \in \{0, 1\}, \quad i \in M \quad (20)$$

$$u_{ir} \geq 0, \forall i \in M, r \in R \quad (21)$$

$$z_{ik} \geq 0, \quad k \in K, i \in M \quad (22)$$

$$\sum_{r \in R} (u_{ir} - u_{jr} + (|V| - 1)x_{ijr}) \leq |V| - 2, \quad \forall i, j \in M \quad (23)$$

$$\sum_{r \in R} u_{ir} \leq |V| - 2, \quad \forall i \in V \quad (24)$$

$$\sum_{s=0}^{j-1} (t_s^{sp} + t_{s,s+1}) = t_j^{in}, \quad \text{where, } t_0^{sp} = 0 \quad (25)$$

$$\left. \begin{array}{l} j^o < t_j^{in} \leq j^c \\ t_j^{ot} = t_j^{in} + t_j^{sp} \\ t_j^{sp} = t_j^p + t_j^l \end{array} \right\} \quad (26)$$

$$t_i^{in} + t_{ijr} x_{ijr} \leq t_j^{in} + H(1 - x_{ijr}), \quad i, j \in M, \quad i \neq j \quad (27)$$

Propose different purchasing costs in different markets be as follows:

$$p_{jk} = \left\{ \begin{array}{ll} p_{jk}^1 - \frac{|p_{jk}^1 - p_{jk}^2|}{\delta} t_j^{in} & : t_j^o < t_j^{in} \leq t_j^o + \delta \\ p_{jk}^2 - \frac{|p_{jk}^2 - p_{jk}^3|}{\delta} t_j^{in} & : t_j^o + \delta < t_j^{in} \leq t_j^o + 2\delta \\ p_{jk}^3 - \frac{|p_{jk}^3 - p_{jk}^4|}{\delta} t_j^{in} & : t_j^o + 2\delta < t_j^{in} \leq t_j^c \end{array} \right. \quad (28)$$

where p_{jk} represents the different purchasing prices at the j th market in different time slots for the k th item. The interval time is divided into three equal sub-slots (δ unit) between opening (t_j^o) and closing (t_j^c). Assume the product's opening and closing prices are p_{jk}^1 and p_{jk}^4 , respectively (Here, $p_{jk}^1 > p_{jk}^2 > p_{jk}^3 > p_{jk}^4$). The product prices are derived for each sub-slot from Eq. (28).

Also, consider different availability in different markets and times, which is defined as follows:

$$q_{jk} = \left\{ \begin{array}{ll} q_{jk}^1 + \frac{|q_{jk}^1 - q_{jk}^2|}{\delta} t_j^{in} & : t_j^o < t_j^{in} \leq t_j^o + \delta \\ q_{jk}^2 - \frac{|q_{jk}^2 - q_{jk}^3|}{\delta} t_j^{in} & : t_j^o + \delta < t_j^{in} \leq t_j^o + 2\delta \\ q_{jk}^3 - \frac{|q_{jk}^3 - q_{jk}^4|}{\delta} t_j^{in} & : t_j^o + 2\delta < t_j^{in} \leq t_j^c \end{array} \right. \quad (29)$$

where q_{jk} represents the availability at the j th market in different time slots for the k th item. As before, the interval time is divided into three equal sub-slots (δ unit) between opening (t_j^o) and closing (t_j^c). Assume the product's opening and closing availabilities are q_{jk}^1 and q_{jk}^4 , respectively (Here, $q_{jk}^1 < q_{jk}^2 > q_{jk}^3 > q_{jk}^4$). The product availabilities are derived from Eq. (29) for each sub-slot.

In this investigation, Z gives the minimum of all system costs (traveling, purchasing, fixed, loading, parking) determined in Eq. (11). Constraints (13) implies that demand for each product is satisfied. Constraints (14) ensures that in the i th market, the purchased amount of product, k cannot exceed the corresponding availability. Constraints (15) and (16) ensure that the purchaser enters and leaves each visited market exactly once. Constraints (17) and (18) guarantee that the tour starts and ends at the depot node. Finally, constraints (19)–(22) are the binary and non-negativity conditions of the decision variables. The sub-tour elimination is defined by the Constraints (23) and (24). Constraints (25) determine the purchaser arrival times at markets. In Eq. (26), j^o and j^c indicate the opening and closing time of the j th market, respectively. t_j^{in} and t_j^{ot} are the in and out time of the purchaser from j th market. t_j^{sp} indicates the total spent time of the purchaser at the j th market, where t_j^p and t_j^l are the purchasing and loading time at the j th market. Constraints (27) arrival times and impose that if arc (i, j) is traversed (i.e. $x_{ijr} = 1$), then $t_j \geq t_i + t_{ijr}$, whereas if $x_{ijr} = 0$, the constraint is always satisfied given a sufficiently large constant H measuring an upper bound on the total time required by the optimal tour (Angelelli et al., 2017). Also, the total time incurred by the system is presented in Eq. (12).

3.5.2. Mathematical formulation of model B, C, D

Model B : Min Z subject to, $T < T_{max}$ with Eqs. (12)–(29)

Model C : Min T with Eqs. (11), (13)–(29)

Model D : Min T subject to, $Z < Z_{max}$ with Eqs. (11), (13)–(29) (30)

4. Proposed multi parent, varied-offspring, variable length quantum-inspired genetic algorithm (MPVOVLQiGA)

Here, we propose a quantum-inspired GA (QiGA) that uses quantum selection, multi parent varied-offspring crossover, and quantum mutation. The detailed process of the proposed MPVOVLQiGA is represented below.

4.1. Quantum concept for TPP

Information is stored in quantum bits (qubits) in the simplest version of quantum computing (Han & Kim, 2002). A quantum qubit

can be a superposition of states 1, 0, or both. A qubit's state can be represented as (Talbi, Draa, & Batouche, 2004):

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (31)$$

where $|0\rangle$ and $|1\rangle$ represent the classical bit values 0 and 1, respectively, with α and β complex numbers such that

$$|\alpha|^2 + |\beta|^2 = 1 \quad (32)$$

The probability values of the qubit in states $|0\rangle$ and $|1\rangle$, respectively, are $||\alpha||^2$ and $||\beta||^2$. A quantum register with n qubits can represent 2^n distinct values in traditional quantum computing. The superposition is destroyed, and only one value is used when the "measure" is taken into account. The exponential increase of the state space with particle number suggested a theoretical exponential speedup of computing on quantum computers compared to classical computers.

In TPP, classical quantum mechanics must be adapted to quantum-inspired techniques, here α and β are generated dependent on the total cost of the particular path (traveling + purchasing + loading + fixed cost + parking) between any two markets i and j with available multipaths between the markets, where, $i, j \in M$ and $r \in R$. For an $|M| = n$ markets TPP, consider an $n \times n$ cost/distance matrix, purchase amount Z_{ik} and purchasing price p_{ik} of each market respectively, where, $i \in M, k \in K$. Considering the overall cost of the particular path with purchasing (traveling+purchasing+loading+fixed cost+parking). Combine them in a single function (Eq. (33)). C_{ijr} is the traveling cost from the i th to the j th market using r th route and $p_{jk}z_{jk}$ represents the purchasing cost at the j th market. To compute α_{ijr} using

$$\alpha_{ijr} = \mu * \frac{c_{ijr} + \sum_{k=1}^K [p_{jk}z_{jk} + z_{ik}*\lambda_{ik} + (\alpha_{ik} + \beta_{ik})*z_{ik}*\gamma_{ik}] + \phi_{ijr}}{\sum_{k=1}^K z_{jk}} \quad (33)$$

where, $i, j = 1, 2, \dots, n$; $k = 1, 2, \dots, K$.

When the value of α_{ijr} has been obtained, the value of β_{ijr} is obtained using Eq. (32). Thus, obtain a quantum representation of the TPP with each state represented in two qubits by an $n \times n$ matrix. For simplicity, μ is taken as 0.001 throughout the work.

4.2. Initialization, representation (variable length chromosome) and path selection

Chromosomes of varied lengths are necessary for solving the TPP. Therefore, initialization in GA with various chromosome lengths is carried out. For clarity, consider the VLGA TPP initialization example in Fig. 9. Consider the following parameters: demand = 100, markets = 10, population = 5, multipath i.e. the number of possible alternate routes between any two marketplaces = 3, and Type-2 Fuzzy logic-based velocity prediction using IoT. Where, 0(2)[0][0] ^{35.00} 4(0)[45] [17] means the purchaser starts from the depot ('0') with 0 purchase items and 0 purchasing price (as depot). He/She chooses the 3rd path ('2') among the three alternate paths (0, 1, 2) with Type-2 Fuzzy logic-based velocity (35 km/h) prediction using IoT. After purchasing the 45 kg in the purchasing cost per unit weight, INR 17 in market 4 moves through the 1st path and so on.

As we consider, multipath in routing i.e. more than one path available between arbitrary two markets. So appropriate path selection perform through maximum β_{jj+1r} , as max β_{jj+1r} determines lower cost (traveling + purchasing) of the path between j th to $(j + 1)^{th}$ market using r th path.

4.3. Quantum selection

We compute a Total β^2 value by taking into account the markets along a path. In addition, predefine a threshold value of β^2 to determine solutions for the mating pool, as β^2 is evaluated based on cost. The mating pool in Algorithm 2 is generated as follows:

Algorithm 1: QUANTUM PATH SELECTION

```

Input: Set of chromosomes
Output: Path selection
1 for  $i \leftarrow 1$  to  $pop - size$  do
2   for  $j \leftarrow 1$  to  $path - size$  do
3     for  $r \leftarrow 1$  to  $R_{jj+1}$  do
4       Evaluate max  $\beta_{jj+1r}^2$  and choose corresponding r

```

Algorithm 2: QUANTUM SELECTION

```

Input: Set of chromosomes
Output: Selective chromosomes
1 for  $i \leftarrow 1$  to  $pop - size$  do
2    $Sum\beta_i^2 = 0$ 
3   for  $j \leftarrow 1$  to  $path - size$  do
4     for  $r \leftarrow 1$  to  $R_{jj+1}$  do
5       Evaluate max  $\beta_{ijr}^2$ 
6        $Sum\beta_i^2 = \beta_i^2 + \beta_{ijr}^2$ 
7    $Total\beta^2 = Sum\beta^2 / (Path - size)$ 
8 for  $i \leftarrow 1$  to  $pop - size$  do
9   if ( $Total\beta^2 > threshold value of \beta^2$ ) then
10    Corresponding chromosome is selected for mating pool
11   else
12    Choose the maximum  $\beta^2$  valued chromosome for
         mating pool

```

4.4. Multi parent varied-offspring crossover

In Fig. 10 represents how a crossover is performed.

Two parent crossover: In this two-parent crossover, the following cases are observed. (i) No child, (ii) One child, (iii) Multiple children (maximum 2, say) In the case of 'No children,' the IVF, i.e., multi parent crossover mechanics, is applied. Due to the application of IVF, there may be (i) No child, (ii) One child, and (iii) Two children. If IVF fails to produce a child, the adoption of one/two children is considered. This is illustrated in Fig. 10.

Multi parent crossover: In real life from parents, a number of children are born randomly. Using multi-offspring may be children are generated 0, 1, 2 (consider up to 2 children maximum). So replace by parent if two children are generated, otherwise to make population size fixed replaced by their parents.

Selection of parents: Number of Parents (NOP) = $(p_c) * (\text{Number of the total population})$, where p_c is the probability of crossover (say 0.4).

Two parent crossover with varied offsprings: On the basis of the above idea, first, we construct the Two-parent crossover mechanism with two parents in the following way.

First of all, two parents are selected randomly from the mating pool, based on the random number between [0, 1]. Select the first parent (say P_1) according to $r < p_c$. Similarly, another parent (say P_2) is selected. The procedure to produce offspring in Two parent Crossover with an example based on five markets TPP is as follows (shown in Fig. 11(a)):

Algorithm of two parent crossover with varied-offspring is presented in Algorithm 3.

Multi parent crossover operations:

On the basis of the above idea, we construct the crossover mechanism with three-parent in the following way:

Initially, three individuals (parents) are selected randomly from the mating pool, based on the random number between [0, 1]. Select the first parent (say P_1) according to $r < p_c$. Similarly, other two parents (say P_2 and P_3) are selected. Details process is shown in Fig. 11(b).

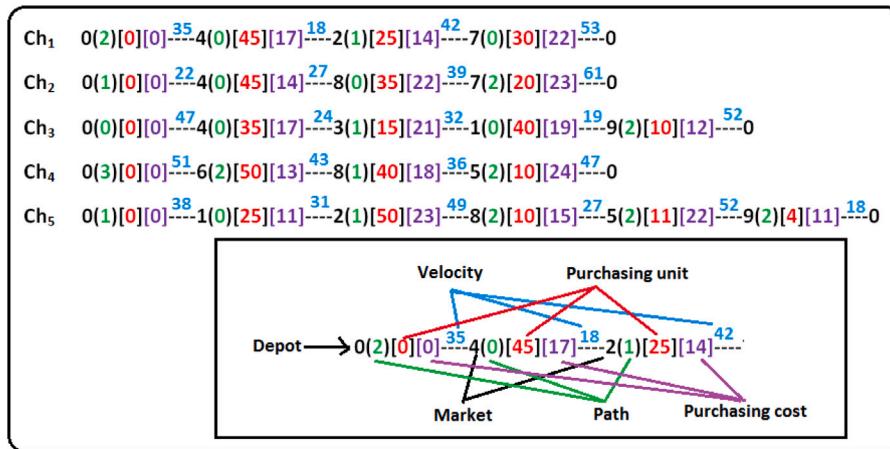


Fig. 9. Graphical representation of Variable length chromosome.

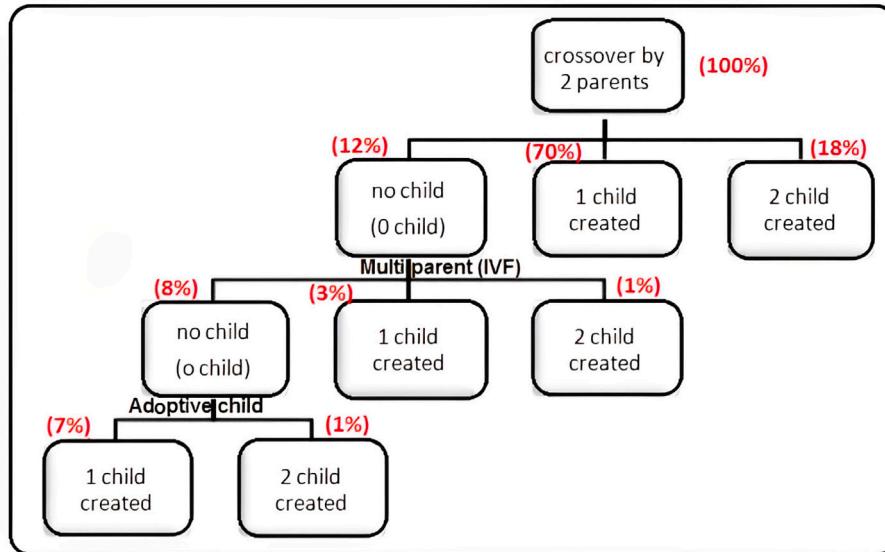


Fig. 10. Overview of crossover.

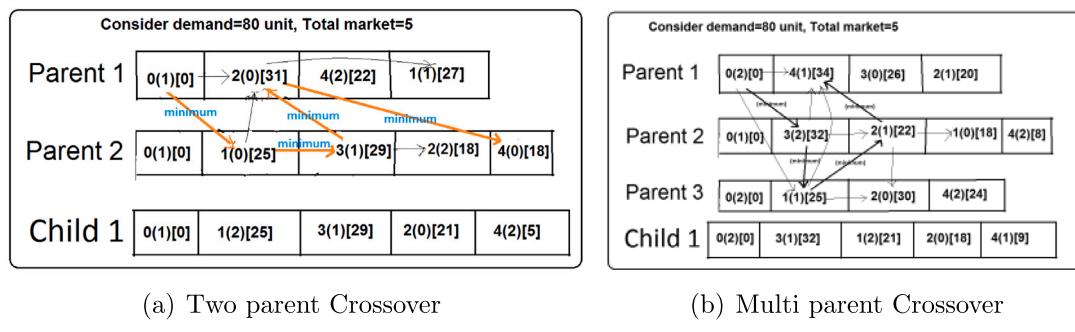


Fig. 11. Schematic view of Crossover.

Detailed operations of multi parent crossover algorithm are given in Algorithm 4.

4.5. Quantum mutation

(i) probability of mutation (p_m), based on current and total generation number is evaluated as follows

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

(ii) mutation process: If $r < p_m$, $r \in \text{rand}[0, 1]$, then the corresponding chromosome is selected for mutation. Now generate three non-repeated random numbers (other than the depot) within the path length of the chromosome. Now calculate β values among the selected markets. Based on the largest value of β , two markets are selected

Algorithm 3: TWO PARENT CROSSOVER WITH VARIED-OFFSPRING

Input: Parents (Two chromosomes)
Output: Child (0, 1, 2 chromosomes)

- 1 Number of parents (NOP) selected for mating pool = $(p_c)^*(\text{pop-size})$
- 2 Total number of crossover (TNC) will be = NOP/2
- 3 **for** $i \leftarrow 1$ **to** TNC **do**
- 4 Choose randomly two distinct parents from the mating pool
- 5 Generate random offspring number (RON)=rand[0, 2]
- 6 Initialize randomly two-parent (P_1, P_2) depends on probability of crossover p_c
- 7 **for** $j \leftarrow 1$ **to** RON **do**
- 8 The child initiates the route with market a_i
- 9 Find the minimum cost (traveling+fixed) from a_i to the next visited market among the two parents
- 10 a_i is updated corresponding to this value
- 11 Repeat Steps 9 and 10 until the demand is fulfilled

Algorithm 4: MULTI PARENT CROSSOVER WITH VARIED-OFFSPRINGS

Input: Parents (Three chromosomes)
Output: Child (0, 1, 2 Chromosome)

- 1 Number of parents (NOP) selected for mating pool = $(p_c)^*(\text{pop-size})$
- 2 Total number of crossover (TNC) will be = NOP/3
- 3 **for** $i \leftarrow 1$ **to** TNC **do**
- 4 Choose randomly three distinct parents from the mating pool
- 5 Generate random offspring number (RON)=rand[0, 2]
- 6 Initialize randomly three-parent (P_1, P_2, P_3) depends on probability of crossover p_c
- 7 **for** $j \leftarrow 1$ **to** RON **do**
- 8 The child initiates the route with market a_i
- 9 Find the minimum cost (traveling+fixed) from a_i to the next visited market among the three parents
- 10 a_i is updated corresponding to this value
- 11 Repeat Steps 9 and 10 until the demand is fulfilled

and swapped (Consider only feasible path as a time-dependent market structure) (see Fig. 12).

Algorithm 5: QUANTUM MUTATION

Input: Selected chromosome
Output: Mutated chromosome

- 1 set g=current generation number
- 2 $p_m = \frac{k}{g}$, $k \in (0, 1)$
- 3 **for** $i \leftarrow 1$ **to** pop - size **do**
- 4 r=rand(0,1)
- 5 **if** ($r < p_m$) **then**
- 6 Choose current chromosome
- 7 Generate non-repeated three random numbers/markets between [2,path-size]
- 8 Evaluate β values among those markets
- 9 Choose the smallest value of β corresponding two markets
- 10 Swap them

A combination of the above steps leads to the proposed MPVOVLQiGA algorithm presented in Algorithm 6.

Algorithm 6: MULTI PARENT, VARIED-OFFSPRING, VARIABLE LENGTH QUANTUM-INSPIRED GENETIC ALGORITHM (MPVOVLQiGA)

Input: Max Gen, Population Size (pop_size), Crossover Probability (p_c), Max Seed, Problem Data (cost, availability, demand, market operating period, and distance matrices)

Output: Set of optimum solutions

- 1 Set initialization s $\leftarrow 1$
- 2 **while** ($s \leq \text{Max seed}$) **do**
- 3 Initialization perform through using sub section Section 4.2
- 4 Quantum path selection done using Algorithm 1
- 5 **while** ($t \leq \text{Max Gen}$) **do**
- 6 Evaluate the fitness of each solution of the initial population
- 7 Quantum selection procedure perform using Algorithm 2
- 8 Crossover procedure through Algorithm 4
- 9 Quantum mutation according to Algorithm 5
- 10 Store the new offspring into offspring set
- 11 Compare the fitness and store the generation best solutions
- 12 $t \leftarrow t + 1$
- 13 Compare the fitness and store seed best solutions
- 14 $s \leftarrow s + 1$
- 15 (Optimum Solution) Store the global optimum and near optimum values

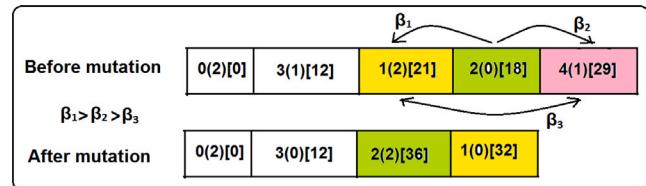


Fig. 12. Overview of mutation.

4.6. Complexity analysis

The complexity of GA for TSP depends on the generation number (G), pop size (M), and number of nodes (N). The proposed MPVOVLQiGA has the $O(GMN^2)$ computational time complexity. Again, in MPVOVLQiGA, population size (MN) is fixed. So it requires fixed space to save the population. Hence space complexity of MPVOVLQiGA is $O(MN)$.

5. Illustration of algorithm and model

5.1. Validation of the algorithm

5.1.1. Test results for MPVOVLQiGA

Solving 15 common benchmark problems from TSPLIB's (Krömer & Uher, 2022), which are given in Table 4, allows us to determine how well the proposed MPVOVLQiGA performs. In terms of the total cost, iterations, and CPU time, compare all the problems. Three separate algorithms (c.f. Table 3)- MPVOVLQiGA, SGA and iGA (Maity, Roy, & Maity, 2015) are used to take 100 independent runs to provide all the findings. The best-known solution is denoted by "BKS" in Table 4, while the best found solution is denoted by "BFS". According to Table 4, MPVOVLQiGA performs better compared to SGA and iGA Maity et al. (2015) in terms of BFS, number of iterations, and CPU time. The size of the problems was reflected in the increase in running time. By

Table 3
Notation and description MPVOVLQiGA, SGA and iGA.

Notation	Description
MPVOVLQiGA	Quantum selection, Multi parent varied-offspring crossover, mutation
SGA	Roulette wheel selection, Cyclic crossover, Random mutation
iGA (Maity et al., 2015)	Probabilistic selection, Comparison crossover, GD mutation

Table 4
Results for Standard TSP Problem (TSPLIB).

Instances	BKS			MPVOVLQiGA			SGA			iGA		
	BFS	Iteration	Time	BFS	Iteration	Time	BFS	Iteration	Time	BFS	Iteration	Time
us16	6859	6859	53	0.07	6859	71	0.09	6859	68	0.07		
gr17	2085	2085	69	0.08	2085	89	0.18	2085	79	0.16		
gr21	2707	2707	158	0.12	2707	161	0.23	2713	181	0.20		
bays29	2020	2020	129	0.21	2020	172	0.58	2028	147	0.32		
dantzig42	699	699	238	0.41	704	279	0.91	720	283	0.49		
eil51	426	426	297	0.72	429	298	1.53	442	294	0.93		
berlin52	7542	7545	337	0.92	7562	409	1.71	7649	387	0.99		
st70	675	675	449	1.43	675	516	2.61	702	489	1.43		
eil76	538	538	497	1.87	544	561	2.95	582	516	2.12		
rat99	1211	1212	756	2.61	1281	881	3.62	1261	799	3.08		
kroA100	21 282	21 310	787	3.82	22 432	1261	4.93	23 717	1256	4.56		
kroA150	26 524	27 283	1154	5.13	28 513	1483	7.42	29 906	1387	6.13		
kroB200	29 437	30 348	1323	5.87	31 316	1632	8.88	34 223	1432	7.26		
a280	2579	2636	1447	7.29	2714	1817	10.34	2956	1647	9.47		
lin318	42 029	43 892	1948	8.61	45 368	2019	11.52	49 931	1997	10.36		

Table 5
Number of wins using different algorithms.

Problem/Algorithm	bayg29	korA100	korB100	korC100	korD100	korE100	korA150	korB150	korA200	korB200	p654	Mean
MPVOVLQiGA	92	89	86	69	84	79	88	91	84	85	83	$\bar{X}_1 = 87.12$
SGA	59	71	48	51	58	46	62	56	53	65	45	$\bar{X}_2 = 58.36$
iGA	72	82	59	76	64	71	65	66	65	62	72	$\bar{X}_3 = 74.28$

Table 6
Analysis of variance.

Source of variation	Sum of square	df	Mean of square	F
Between groups	$SS_B = 4548.23$	J-1=2	$MS_B = \frac{SS_B}{J-1} = 2274.118$	
Within groups	$SS_W = 13.62$	J(I-1)=30	$MS_W = \frac{SS_W}{J(I-1)} = 45.42$	$\frac{MS_B}{MS_W} = 50.06$
Total	$SS_T = 5910.83$	IJ-1=32		

MPVOVLQiGA, us16, gr17, gr21, bays29, dantzig42, eil51, st70, eil76, these problems are achieved with optimal solutions.

5.1.2. Statistical test

Analysis of variance (ANOVA), was performed using benchmark instances as reported in Table 4 to judge the efficiency of three algorithms: MPVOVLQiGA, simple GA (SGA) (roulette wheel (RW) selection, cyclic crossover, random mutation), and an improved GA (iGA) (probabilistic selection, comparison crossover, generation dependent mutation) proposed by Maity et al. (2015). The number of successes in 100 distinct runs for the specified benchmark instances using the algorithms MPVOVLQiGA, SGA, and iGA, respectively, is shown in Table 5.

For each algorithm ($J=3$), there were an equal number of benchmark examples ($I=11$). The mean of the sample mean is $\bar{X} = 73.25$.

The critical F -value is $F_{0.05(2,30)} \approx 3.32$. Because the computed F value (in Table 6) is larger (50.06) than the critical F value with 55% level of significance, we conclude that the performance attained by the heuristic is statistically different. When there are more than two groups in an ANOVA, the F value is significant. Scheffe's multiple comparisons F -test is then used to evaluate which group means are substantially different. The computed F value for the first pair, MPVOVLQiGA, and SGA, is derived by $F = \frac{(\bar{X}_1 - \bar{X}_3)^2}{MS_W(\frac{1}{J} + \frac{1}{I})} = 13.29$. The estimated F value for the second pair, MPVOVLQiGA, and iGA, is 42.78. The estimated F values for MPVOVLQiGA and iGA and MPVOVLQiGA and SGA differ

significantly from each other since both are more significant than the crucial value (3.32). In Table 5, according to reports, the mean (\bar{X}_1) of X_1 is greater than the other two means, \bar{X}_2 and \bar{X}_3 . Significant differences exist between the algorithms, so we conclude that MPVOVLQiGA works better than the other two.

For the above tests, the following computer configuration and parametric values are used.

5.2. System configuration for implementation

- (1) System: Windows 2010
- (2) CPU: CORE i5
- (3) RAM: 4 GB
- (4) Software: Code Block

The parameter values used in the experiments are in given Table 7.

5.3. Real-life experiment

We implement the proposed model in the BANKURA district, West Bengal, India (cf. Fig. 13). There is a depot, Bankura town (marked by '0'), and 19 different markets ((1) Beliatore, (2) Ramharipur, (3) Panchal, (4) Nikunjapur, (5) Mankanali, (6) Sunukpahari, (8) Onda, (9) Gopalpur, (10) Jhantipahari, (11) Jaganathpur, (12) Kamalpur, (13) Manikbazar, (14) Sonamukhi, (15) Harigram, (16) Molian, (17) Ramsagar, (18) Saldiha, (19) Bheduasole). Thus IoT-T2FL-MPTPPswTDMSSs are formulated following the TPP procedure with 20 markets and 3

Table 7

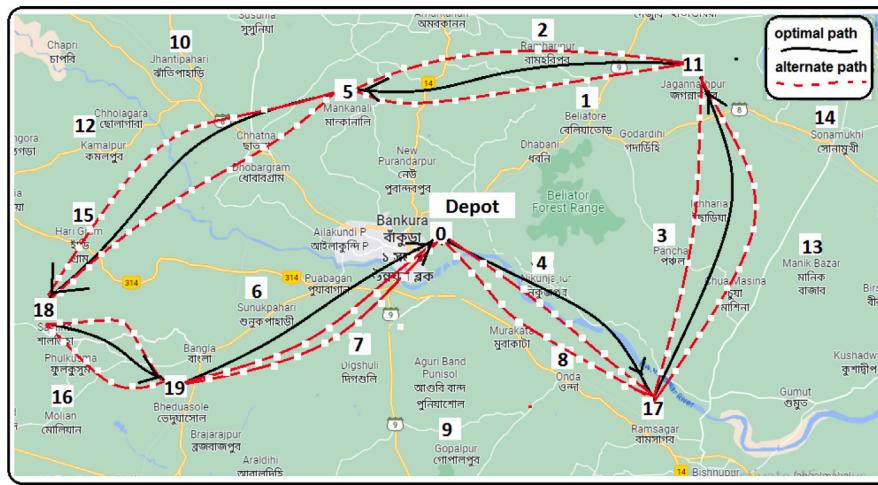
Parameters values chosen for validation of the algorithm.

Parameters	Domain value/Range	Parameters	Domain value/Range
Number of Chromosome	100	p_c	0.36
Max_seed	100	p_m	0.12
Max_Generation	2000	Qubit threshold	0.65

Table 8

Input parameters for the model experiment.

Parameters	Value	Parameters	Value
Number of markets (N)	20	z_{ik}	10–50 kg
Predicted velocity (P_{vel})	15–65 km/h	p_{ik}	11–25 INR
Loading time	(Purchasing amount)/100	q_{ik}	10–50 kg
Purchasing time	(Purchasing amount)/50		
Loading cost (INR)	(Purchasing amount)*rand(1.5, 2.5)		
Parking charge	(Loading time)*rand(80, 120)		

**Fig. 13.** Optimal path for Purchaser.

alternative paths between every two markets (cf. Section 3.5.1), each with its own set of distances, traveling costs matrices. The values (a, b, c) represent the first, second, and third paths, respectively. The path structure is shown in Fig. 13.

5.3.1. Input data

The distance, traveling cost per unit distance, and fixed charge cost for

IoT-T2FL-MPTPPswTDMS are presented in https://github.com/somnathmajivucs/Input_matrix_tpp and Table 8.

For the distance and traveling cost (a,b,c)(represented by $\{0,1,2\}$) (say), the values a, b and c are for the 1st, 2nd and 3rd paths respectively. Market operating period, availability, and purchase price range of products are given in Table B.2.

5.3.2. Optimum results of IoT-T2FL-MPTPPswTDMS:

Using the above-mentioned data (Tables are presented in https://github.com/somnathmajivucs/Input_matrix_tpp, B.2), Models-(1, 2, 3) are solved by MPVOVLQiGA and presented in Tables 9–15 and Figs. 14–17.

Optimum results of Model A (Cost minimization):

Tables 9 represents the overall scenario of Model A. In Table 9, the optimal multipath is $0(0)[0]^{47.69} 17(1)[49]^{53.74} 11(0)[40]^{37.19} 5(2)[41]^{46.37} 18(2)[43]^{49.25} 19(2)[27]^{37.29} 0$. The traveling cost, purchasing cost, loading cost, parking cost, fixed cost, total cost, travel time, purchasing time, loading time, and total time are INR 386, INR 3125, INR 426, INR 575, INR 90, INR 4602, 4.93 h., 4 h., 2 h., and 10.93 h. respectively.

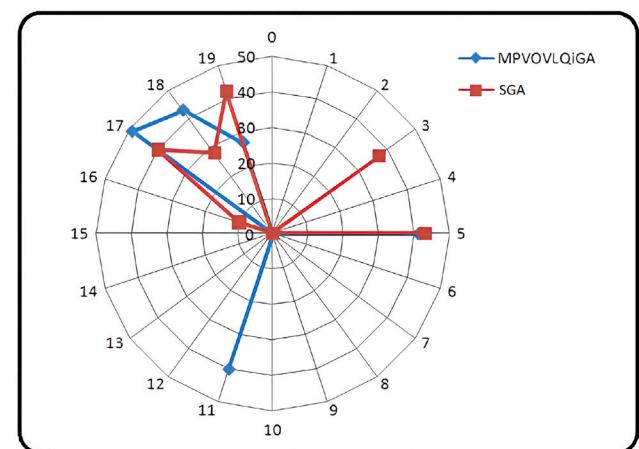
**Fig. 14.** Result with different purchased units in different markets.

Fig. 14 represents the different purchasing amounts in different markets for Model A.

In Tables 10 and 11, and Fig. 15 present the different optimum results when demands are different.

Table 12, and Fig. 16 furnish the different optimum results when journey starting time and demands of the purchaser is different. It is observed that when the demand is 250 units and starting time is up to

Table 9

Results of time-dependent availability and purchasing cost of IoT-T2FL-MPTPPswTDMS.

Algorithm	Demand	Market[Travel path][Purchase Units]										Visited markets	Traveling Cost(INR)	Purchasing Cost(INR)	Loading Cost(INR)	Parking Cost(INR)	Fixed Cost(INR)	Total Cost(INR)	Travel Time (h)	Purchasing Time (h)	Loading Time (h)	Total Time (h)
PA	MP	200	0(0) [0] $\frac{47.69}{46.37}$ 17(1) [49] $\frac{53.74}{49.25}$ 11(0) [40] $\frac{37.19}{37.29}$ 5(2) [41] 18(2) [43] $\frac{49.25}{41.93}$ 19(2) [27] $\frac{37.29}{55.27}$ 0	6	386	3125	426	575	90	4602	4.93	4	2	10.93								
	2D	200	0(0) [0] $\frac{48.29}{41.93}$ 14(0) [41] $\frac{56.32}{55.27}$ 2(0) [41] $\frac{39.27}{41.26}$ 17(0) [35] 15(0) [20] $\frac{41.26}{58.24}$ 6(0) [17] $\frac{58.24}{0}$	7	507	3212	390	611	142	4862	5.21	4	2	11.21								
SGA	MP	200	0(0) [0] $\frac{39.82}{35.67}$ 19(2) [42] $\frac{54.38}{39.84}$ 3(2) [37] $\frac{47.28}{46.32}$ 17(0) [40] 5(2) [43] $\frac{39.84}{48.47}$ 16(0) [10] $\frac{46.32}{57.24}$ 18(0) [28] $\frac{57.69}{0}$	7	573	3210	411	591	139	4914	5.31	4	2	11.31								
	2D	200	0(0) [0] $\frac{41.39}{54.12}$ 2(0) [35] $\frac{57.24}{47.34}$ 19(0) [40] $\frac{49.38}{40.27}$ 7(0) [33] 15(0) [45] $\frac{47.34}{51.37}$ 4(0) [27] $\frac{51.37}{44.39}$ 16(0) [10] $\frac{44.39}{0}$	7	672	3410	435	676	158	5341	6.2	4	2	12.2								

PA: MPVOVLQiGA (proposed algorithm), MP: multipath

Table 10

Results of Model A with different demand.

Algorithm	Demand	Market[Travel path][Purchase Units]										Visited markets	Traveling Cost(INR)	Purchasing Cost(INR)	Loading Cost(INR)	Parking Cost(INR)	Fixed Cost(INR)	Total Cost(INR)	Travel Time (h)	Purchasing Time (h)	Loading Time (h)	Total Time (h)
PA	MP	100	0(0) [0] $\frac{46.38}{55.24}$ 16(0) [45] $\frac{39.24}{12(2)}$ 7(2) [44] [11] $\frac{47.38}{0}$	4	372	2099	224	338	102	3135	4.06	2	1	7.06								
		150	0(0) [0] $\frac{37.45}{48.47}$ 9(0) [45] $\frac{46.82}{12(0)}$ 6(1) [45] $\frac{54.37}{13(1)}$ 14(0) [47] $\frac{46.35}{0}$	5	396	2443	315	486	108	3748	4.26	3.0	1.5	8.76								
	2D	200	0(0) [0] $\frac{47.69}{46.37}$ 17(1) [49] $\frac{53.74}{47.34}$ 11(0) [40] $\frac{37.19}{37.29}$ 5(2) [41] 18(2) [43] $\frac{49.25}{40.27}$ 19(2) [27] $\frac{37.29}{0}$	6	386	3125	426	575	90	4602	4.93	4	2	10.93								
		250	0(0) [0] $\frac{53.74}{45.32}$ 9(2) [49] $\frac{47.29}{40(4)}$ 15(1) [39] $\frac{33.47}{37.59}$ 6(1) [40] $\frac{45.63}{2(1)}$ 17(2) [34] $\frac{47.86}{0}$	7	548	3225	489	723	179	5164	5.21	4	3.5	12.71								

PA: MPVOVLQiGA (proposed algorithm), MP: multipath

Table 11

Results of IoT-T2FL-MPTPPswTDMS with different demand.

Demand	MPVOVLQiGA		GA	
	Multipath	Single path	Multipath	Single path
250	5164	no feasible path	5857	no feasible path
225	4786	5128	5085	5558
200	4602	4862	4914	5341
175	4246	4584	4627	4958
150	3748	4354	4297	4625
125	3389	3974	4027	4347
100	3135	3597	3456	3742

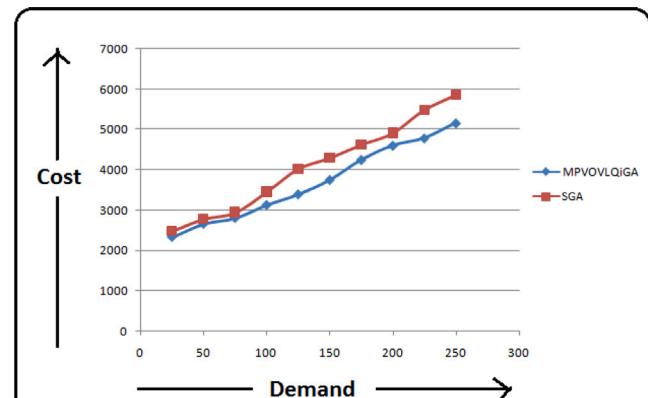
08:00, only a feasible path is available; starting time after 09:00 with a demand of 250 cannot find feasible solutions. Similar scenarios are available in [Table 12](#) for different demands and starting times.

Optimum results of Model B (Cost minimization with a time constraint):

[Table 13](#) represents the total system cost with different time constraints. It is seen that when the time limit decreases, the system cost increases ([Fig. 17](#)), but when the time limit is less than 10.5 h., it is unable to find the feasible path in the system. Again when the time limit increases, the system cost is minimized to a certain level. The system cost remains unaltered when the time limit exceeds 11.0 h. ([Table 13](#)). This behavior is as per expectation.

Optimum results of Model C (Time minimization):

In [Table 14](#), the optimum routing path is 0(0)[1] $\frac{46.35}{49(5)}$ 11(2)[50] $\frac{44.70}{42.37}$ 9(0)[49] $\frac{51.28}{3(1)}$ 3(1)[49]

**Fig. 15.** Result with different demand vs cost.

14(2)[49] $\frac{35.67}{0}$. The traveling cost, purchasing cost, loading cost, parking cost, fixed cost, total cost, travel time, purchasing time, loading time, and total time are INR 426, INR 4336, INR 415, INR 752, INR 131, INR 5503, 4.55 h., 4 h., 2 h., and 10.55 h. respectively.

Optimum results of Model D (Time minimization with cost constraints): In [Table 15](#), optimum results are presented when cost constraint is incorporated. Here, with the tightening of the cost constraint, time increase. This is as per expectation and illustrated in [Table 15](#).

Table 12

Results of IoT-T2FL-MPTPPswTDMS with different starting time with different demand.

Demand	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00
250	5164	5287	5024	5189	NFP	NFP	NFP	NFP	NFP	NFP
225	4786	4657	4796	4627	4796	NFP	NFP	NFP	NFP	NFP
200	4602	4758	4569	5072	5437	4674	5627	NFP	NFP	NFP
175	4246	4287	4215	4125	4257	4269	4198	4237	NFP	NFP
150	3748	3736	3812	4067	3957	3857	3976	3812	3961	NFP
125	3389	3734	3448	3716	3659	3748	3567	3641	3426	3666
100	3135	3564	3094	3487	3669	3513	3458	3364	3296	3475

NFP: No feasible path

Table 13

Results of IoT-T2FL-MPTPPswTDMS with time constraint.

Time limit (h)	MPVOVLQiGA		GA	
	Total cost (with multipath)	Total cost (with single path)	Total cost (with multipath)	Total cost (with single path)
11.1	4602 (same)	4735	4947 (same)	5984
11.0	4602 (same)	4946	4947 (same)	5752
10.9	4826	5026	4984	5367
10.8	5042	6369	6279	no feasible path
10.7	5437	no feasible path	7346	no feasible path
10.6	5791	no feasible path	no feasible path	no feasible path
10.5	no feasible path	no feasible path	no feasible path	no feasible path

Table 14

Results of Model C of IoT-T2FL-MPTPPswTDMS.

Algorithm	Demand	Market[Travel path][Purchase Units]	Visited markets	Traveling Cost(INR)	Purchasing Cost(INR)	Loading Cost(INR)	Parking Cost(INR)	Fixed Cost(INR)	Total Cost(INR)	Travel Time (h)	Purchasing Time (h)	Loading Time (h)	Total Time (h)
PA	multipath	200 0(0) [1] ^{46.35} 11(2) [50] ^{44.70} 9(0) [49] ^{51.28} 3(1) [52] ^{42.37} 14(2) [49] ^{35.67} 0	5	426	4336	415	752	131	5503	4.55	4	2	10.55
		200 0(0) [0] ^{41.27} 14(0) [42] ^{49.74} 1(0) [44] ^{38.46} 5(0) [38] ^{47.85} 16(0) [40] ^{53.67} 7(0) [36] ^{45.47} 0											
SGA	MP	200 0(0) [2] ^{31.29} 6(1) [42] ^{52.87} 15(0) [37] ^{47.39} 4(0) [38] ^{46.36} 17(2) [43] ^{44.34} 3(1) [40] ^{55.62} 0	6	618	4669	439	748	148	6322	5.21	4	2	11.21
		200 0(0) [0] ^{38.92} 15(0) [35] ^{46.27} 1(0) [46] ^{54.36} 7(0) [41] ^{48.51} 14(0) [43] ^{44.76} 3(0) [35] ^{31.29} 0											

PA: MPVOVLQiGA (proposed algorithm), MP: multipath

Table 15

Results of IoT-T2FL-MPTPPswTDMS under time minimization with cost constraints.

Cost limit (INR)	MPVOVLQiGA		SGA	
	Total time (with multipath)	Total time (with single path)	Total time (with multipath)	Total time (with single path)
5800	10.51	10.81	10.79	11.86
5600	10.51	10.83	10.81	11.89
5400	10.73	10.96	10.93	12.34
5200	10.81	11.01	11.02	no feasible path
5000	10.86	11.16	11.09	no feasible path
4800	11.03	no feasible path	no feasible path	no feasible path
4600	no feasible path	no feasible path	no feasible path	no feasible path

6. Dissection of the results

6.1. Impact of different time and cost constraints

Depending on the different time cost limits, we evaluate the feasible solutions of Models B and D, and the results are presented in Figs. 18(a) and 18(b). It is observed that for Model B, the time limit below 10.5 h.

is unable to get the feasible solutions. Again, for the time limit above 11 h., the solution remains unaltered.

Similarly for Model D, the cost limit of less than INR 4600 does not generate a feasible solution (Fig. 18(b)). If the cost exceeds INR 5800, the solution remains unchanged. The same pattern of results is observed using both SGA and MPVOVLQiGA. The proposed method, MPVOVLQiGA gives better results than SGA.

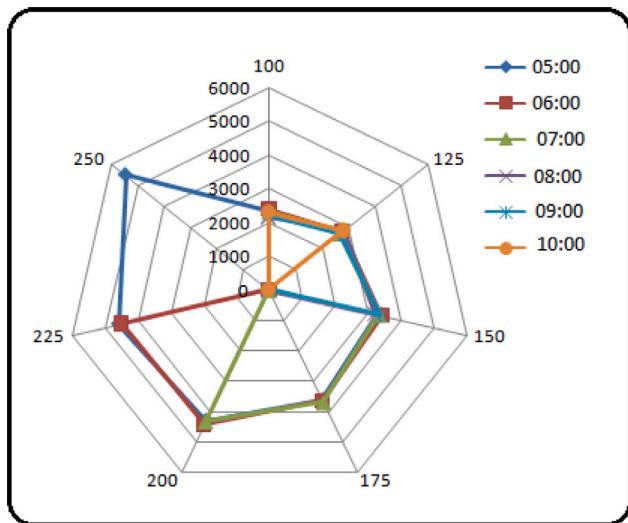


Fig. 16. Result with different demand with different starting time.

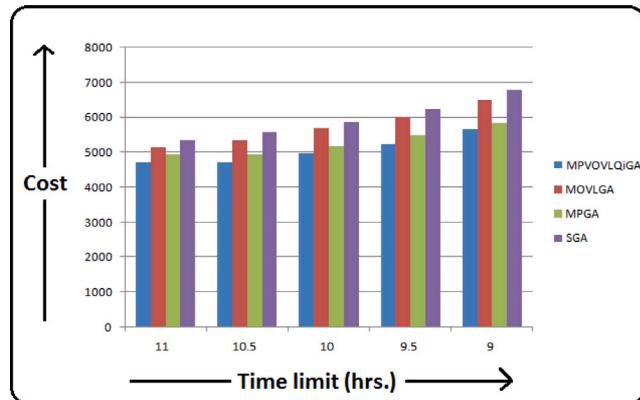


Fig. 17. Result with different time limit.

6.2. Impact of different starting times on cost and time for a fixed demand

Figs. 19(a) and 19(b) furnish for fixed demand (200 units), optimal system costs, and time against different starting times for the purchaser. Based on this, the purchaser can decide the appropriate starting time for minimum cost/time. If it is not possible to start the journey at a particular time, for minimum cost/time, the purchaser can select the starting time as per his/her convenience depending on the available resource. It is to be noted that for optimum costs, the purchaser's starting time should be before 11:15 (by MPVOVLQIGA). However, SGA gives a more conservative time, 09:45, for this.

For the same fixed demand (200 units), Fig. 19(b) gives the optimum (min) times against different starting times. Here, no feasible solution is available after the starting time, 12:42 h. as MPVOVLQIGA (10:28) by SGA.

6.3. Impact of multipath for time and cost optimization

Due to advances in infrastructure throughout the world, including developing countries like India, etc., the usage of multipath in a routing paradigm is more practical. For the first time, the availability of three different connecting paths for travel between two arbitrary nodes is

considered. Multiple-path Models A, B, C, and D outperform the corresponding single-path Models in terms of optimal results (cf. Tables 9–15).

Consider the optimal path of Model C (with multipath) i.e., 0(0) [1] $\frac{46.35}{11}(2)[50]$ $\frac{44.70}{9}(0)[49]$ $\frac{51.28}{3}(1)[52]$ $\frac{42.37}{14}(2)[49]$ $\frac{35.67}{0}$.

Here (in Fig. 20), a travel path segment is $11(2)[50]$ $\frac{44.70}{9}(0)[49]$ means that 3rd path is used among three alternate paths between markets 11 and 9, 50 units are purchased at market 11 and velocity of the vehicle is 44.7 km/hr. The distance between markets is 10 km and the travel time is 0.27 h. As an alternative, if we consider only a single path between 11 and 9 markets, say the first path (0); distance is increased by 1 km, also time increases by 0.04 h., i.e., 0.31 h. For the second path (1), distance is reduced by 1 km, but time increases by 0.01 h., i.e., 0.28 h. by this, it is established that the lowest distance path is not the optimum path. In this case, for minimum time, the path with a 10 km distance is chosen.

Similar behavior is observed for cost minimization also. So multi-path/alternate paths give better routing options for optimization.

6.4. Impact of time-dependent market structure

Here, we compare the results of Models- A and C, dependent and independent of time t (see Fig. 21).

Figs. 21(a) and 21(b) show the performance of cost and time respectively against demand. In both cases (21(a), 21(b)), for the present data, the time-independent market structure (markets open for 24 h) performs better. But markets follow the time-dependent market structure in real life. This can be verified with real-life data.

6.5. Managerial insight

In Table 16, we present the managerial decisions (to opt out of the best results) in question-and-answer forms.

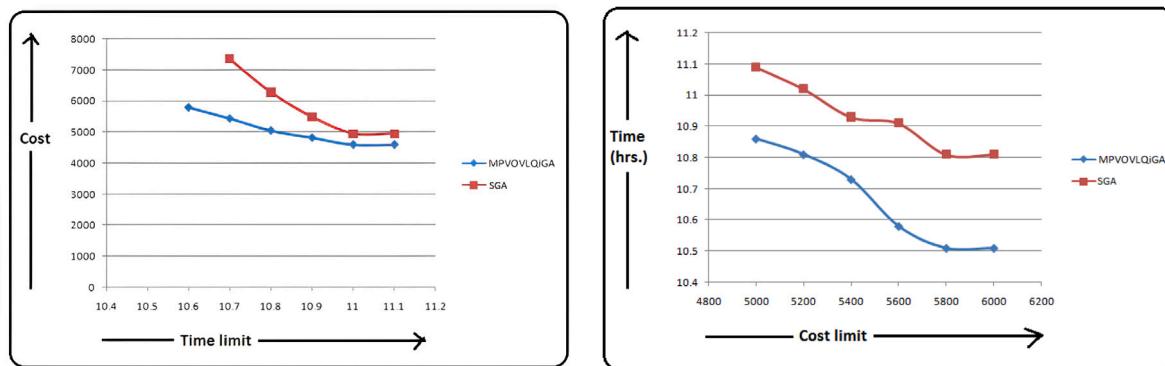
7. Conclusions

In this present study, two fold investigations are done- (i) realistic time-dependent market structure based TPP model using IoTs and Type-2 fuzzy logic and (ii) problem-specific variable length quantum-inspired genetic algorithm.

In the TPP model, we incorporated a time-dependent market structure, i.e., time-dependent availability and purchasing costs of distinctly located markets, along with time and cost constraints. To make the model more realistic, alternate paths are considered between two arbitrary markets, which helps to generate more feasible solutions in the solution space. Here information regarding weather, road surface, and congestion wrt different paths obtained by the IoT is used as inputs in T2FL, and path-wise vehicles' average velocities are obtained. Also, fixed costs due to toll-plaza, etc., are incorporated. With the different demands and time constraints, we observed various scenarios related to the model. Starting at different times, for a particular demand, the optimum costs and times are different. We have derived the minimum cost (time) for a time-dependent market structure with time (cost) constraint against a fixed demand, where availabilities and prices of items change with the operating time of the market.

To solve the above problem, we introduced a multi parent, varied-offspring, variable length quantum-inspired genetic algorithm (MPVOVLQIGA) with quantum selection, multi parent, varied-offspring crossover, and quantum mutation. The variable length concept for TPP is more realistic, as demand may be satisfied without visiting all the markets. The superiority of the intended algorithm is established by testing it on benchmark TPP instances using a statistical test.

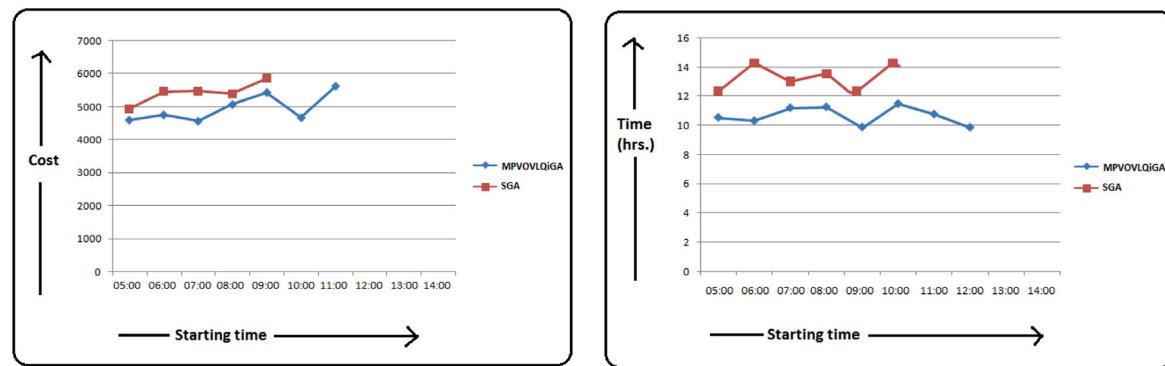
The limitations of this investigation that can be studied further are: (i) In this model, considered single-depot TPP may be extended to a multi-depot problem. (ii) Hypothetical input data (distance, cost, time,



(a) Cost vs time limit

(b) Time vs cost limit

Fig. 18. Cost vs time and cost limit.



(a) Cost vs different starting time

(b) Time vs different starting time

Fig. 19. Different starting time, cost and time.

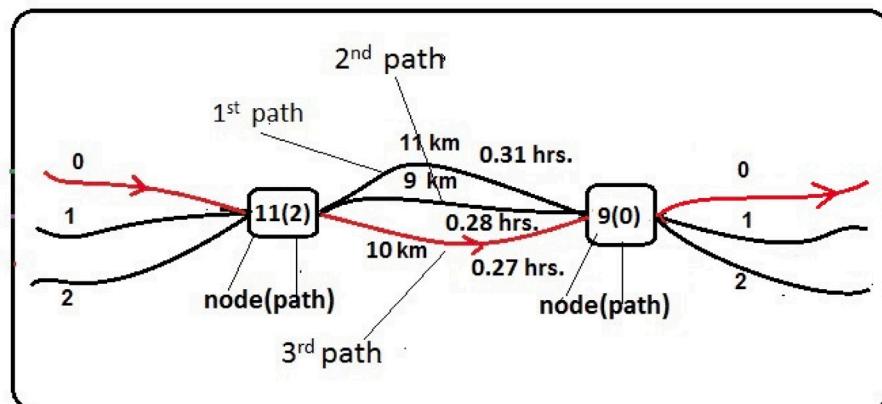
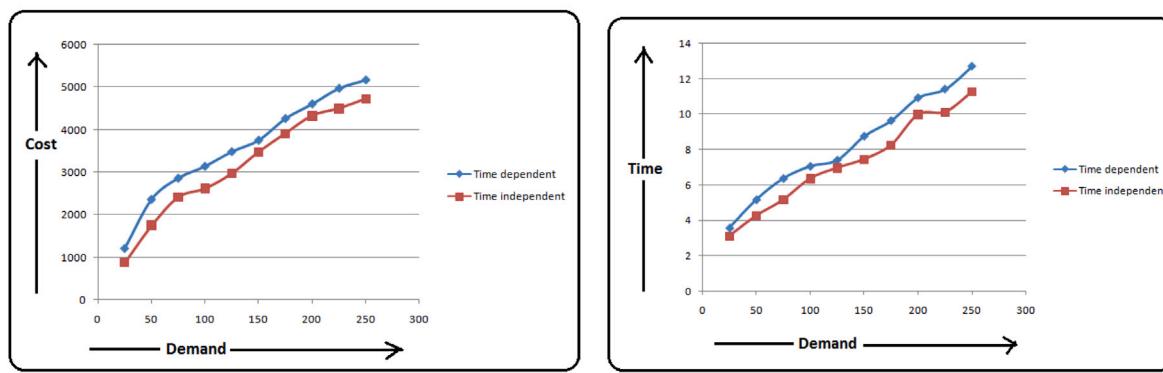


Fig. 20. Justification of multipath.



(a) Demand vs cost

(b) Demand vs time

Fig. 21. Result with different demand, cost and time.

Table 16

Research questions, solutions and managerial insight.

Research questions	Solutions	Managerial insight
Q1: How to choose the markets?	Tables 9, 10 and 14 and Fig. 14	Management can take appropriate decision to choose the market.
Q2: If the price of an item goes down within a time slot, how-to take the advantages of reduced prices?	Tables 9, 10 and 14 and Fig. 14	Management can choose the appropriate time slots for the markets to get the reduced price at the product
Q3: If the items availabilities also vary within the markets' period, how to utilize the maximum availability situations and to make the trade off between the availability and price?	Tables 9, 10 and 14 and Fig. 14	Management should try to buy the maximum product at lower prices and focus on optimal result
Q4: What will be the routing plan of a purchaser if the real-life road conditions (weather, congestion and surface) are considered and monitored by the IoT?	Table 9, Section 3.4 Fig. 13	The velocity is determined based on IoT-Type-2 fuzzy logic. Then traveling time is evaluated for the routing plan
Q5: How to get appropriate travel velocity out of these crisp data, using the deterministic data through Type-2 fuzzy logic (say) involving imprecise data.	Table 9, Section 3.4 Fig. 13	Crisp IoT data are fed to the Type-2 fuzzy logic controller and the travel velocity for each path is obtained as an input
Q6: How to solve these problems using GA?	Table 9, and Fig. 9	For this investigation, we introduce problem-specific VLGA
As demand is fixed and availabilities at markets are different, the length of chromosome in GA will be different.		
Q7: How to formulate a variable length GA with multiple parents and varied off-springs to solve IoT-T2FL-MPTPPswTDMS? Will it be faster if the quantum concept is introduced in the GA?	Table 9, and Fig. 11	A problem-specific variable length GA inspired by quantum mechanics is developed for the solution. This is faster than other algorithms.
Q8: How to address the time dependent market structure along with time and cost constraints?	Table 15 Fig. 18	Different market operating time, a real-life scenario can be handled through these models.
Q9: When should purchaser starts	Table 11	Optimum results (cost

(continued on next page)

Table 16 (continued).

Research questions	Solutions	Managerial insight
his journey and what the procurement plan for satisfying the predefined the demand for the particular product?	Fig. 19	and time) depends on the journey starting time of the purchaser. A purchaser can decide his starting following the present analysis.
Q10: How to manage purchasing process, entire purchasing time/procurement time in an efficient manner?	Table 13 Figs. 18, 19	Considering time dependent availability and purchasing cost, management can take appropriate routing plan.
Q11: How the optimal routing plan considering multipath, purchasing cost, traveling cost along with fixed costs?	Tables 9–15 and Fig. 20	It is illustrated that multipath performs better than single path models.

Table A.1
Fuzzy 25 rules.

Input		Output	
Weather	Road surface	Road congestion	Velocity
very good	very smooth	very rare	very very high
very good	very smooth	rare	very high
very good	very smooth	medium	very high
very good	very smooth	often	high
very good	very smooth	very often	high
good	smooth	very rare	very high
good	smooth	rare	very high
good	smooth	medium	high
good	smooth	often	high
good	smooth	very often	medium
medium	medium	very rare	high
medium	medium	rare	high
medium	medium	medium	medium
medium	medium	often	medium
medium	medium	very often	low
bad	rough	very rare	medium
bad	rough	rare	medium
bad	rough	medium	low
bad	rough	often	low
bad	rough	very often	very low
very bad	very rough	very rare	low
very bad	very rough	rare	low
very bad	very rough	medium	very low
very bad	very rough	often	very low
very bad	very rough	very often	very very low

etc.) are considered. A case study is preferred. (iii) Experiments for the models are performed with 20 markets only. These can be illustrated with a larger market structure.

In the future, the proposed model can also be extended with (i) two vehicles along- one for the purchaser and another for goods with multipaths; (ii) cost and time constraints may be fuzzy (iii) multi-depot TPP, (iv) multiple purchasers and multi-items.

Funding

No funding was received for conducting this study.

Ethics approval

This article does not contain any studies with human participants or animals performed by any authors.

Availability of data and material

All the information is real-life data that we gathered from local and online sources. We also addressed an existing problem for comparison, which we cited and discussed in the article.

CRediT authorship contribution statement

Somnath Maji: Conceptualization, Methodology, Software, Writing – original draft. **Kunal Pradhan:** Conceptualization, Validation, Methodology, Software. **Samir Maity:** Validation, Software, Writing – review & editing, Writing – original draft. **Izabela Ewa Nielsen:** Supervision, Validation, Review – original draft. **Debasis Giri:** Supervision, Validation, Review – original draft. **Manoranjan Maiti:** Conceptualization, Supervision, Validation, Writing – review & editing.

Data availability

Data will be made available on request.

Appendix A. Fuzzy 25 rules

See [Table A.1](#).

Appendix B. Input data

See [Table B.2](#).

Table B.2

Input data: Market parameters during different time slots operating time slot(s).

Market	Parameters	Time slot 1	Time slot 2	Time slot 3
Market-0 (Depot)	Time		(5:00-22:00)	
	Availability		NA	
	Price		NA	
Market-1	Time	(5:00-7:00)	(7:00-11:00)	(11:00-13:00)
	Availability	(40 unit-50 unit)	(50 unit-20 unit)	(20 unit-10 unit)
	Price	(25 INR-22 INR)	(22 INR-18 INR)	(18 INR-15 INR)
Market-2	Time	(8:00-11:00)	(11:00-15:00)	
	Availability	(39 unit-46 unit)	(46 unit-33 unit)	
	Price	(18 INR-19 INR)	(19 INR-22 INR)	
Market-2	Time	(18:00-19:00)	(19:00-22:00)	
	Availability	(34 unit-42 unit)	(42 unit-31 unit)	
	Price	(21 INR-18 INR)	(18 INR-14 INR)	
Market-3	Time	(10:00-12:00)	(12:00-16:00)	(16:00-18:00)
	Availability	(41 unit-52 unit)	(52 unit-23 unit)	(23 unit-12 unit)
	Price	(26 INR-23 INR)	(23 INR-19 INR)	(19 INR-15 INR)
Market-4	Time	(6:00-8:00)	(8:00-12:00)	(12:00-15:00)
	Availability	(38 unit-49 unit)	(49 unit-22 unit)	(22 unit-14 unit)
	Price	(24 INR-21 INR)	(21 INR-16 INR)	(16 INR-14 INR)
Market-5	Time	(5:00-6:00)	(6:00-9:00)	(9:00-11:00)
	Availability	(37 unit-46 unit)	(46 unit-30 unit)	(30 unit-21 unit)
	Price	(23 INR-21 INR)	(21 INR-18 INR)	(18 INR-14 INR)
Market-6	Time	(5:00-7:00)	(7:00-11:00)	
	Availability	(39 unit-46 unit)	(46 unit-32 unit)	
	Price	(22 INR-19 INR)	(19 INR-14 INR)	
Market-6	Time	(17:00-19:00)	(19:00-22:00)	
	Availability	(28 unit-35 unit)	(35 unit-24 unit)	
	Price	(21 INR-18 INR)	(18 INR-14 INR)	
Market-7	Time	(5:00-7:00)	(7:00-11:00)	(11:00-13:00)
	Availability	(34 unit-45 unit)	(45 unit-23 unit)	(23 unit-17 unit)
	Price	(24 INR-21 INR)	(21 INR-19 INR)	(19 INR-16 INR)
Market-8	Time	(11:00-13:00)	(13:00-17:00)	(17:00-20:00)
	Availability	(36 unit-44 unit)	(44 unit-31 unit)	(31 unit-24 unit)
	Price	(25 INR-23 INR)	(23 INR-20 INR)	(20 INR-16 INR)
Market-9	Time	(5:00-7:00)	(7:00-11:00)	
	Availability	(41 unit-50 unit)	(50 unit-32 unit)	
	Price	(24 INR-22 INR)	(22 INR-17 INR)	
Market-10	Time	(18:00-19:00)	(19:00-22:00)	
	Availability	(35 unit-44 unit)	(44 unit-30 unit)	
	Price	(22 INR-20 INR)	(20 INR-16 INR)	
Market-11	Time	(5:00-7:00)	(7:00-11:00)	(11:00-13:00)
	Availability	(40 unit-50 unit)	(50 unit-20 unit)	(20 unit-10 unit)
	Price	(25 INR-22 INR)	(22 INR-18 INR)	(18 INR-15 INR)
Market-12	Time	(8:00-11:00)	(11:00-15:00)	
	Availability	(39 unit-46 unit)	(46 unit-33 unit)	
	Price	(18 INR-19 INR)	(19 INR-22 INR)	
Market-12	Time	(18:00-19:00)	(19:00-22:00)	
	Availability	(34 unit-42 unit)	(42 unit-31 unit)	
	Price	(21 INR-18 INR)	(18 INR-14 INR)	
Market-13	Time	(10:00-12:00)	(12:00-16:00)	(16:00-18:00)
	Availability	(41 unit-52 unit)	(52 unit-23 unit)	(23 unit-12 unit)
	Price	(26 INR-23 INR)	(23 INR-19 INR)	(19 INR-15 INR)
Market-14	Time	(6:00-8:00)	(8:00-12:00)	(12:00-15:00)
	Availability	(38 unit-49 unit)	(49 unit-22 unit)	(22 unit-14 unit)
	Price	(24 INR-21 INR)	(21 INR-16 INR)	(16 INR-14 INR)
Market-15	Time	(5:00-6:00)	(6:00-9:00)	(9:00-11:00)
	Availability	(37 unit-46 unit)	(46 unit-30 unit)	(30 unit-21 unit)
	Price	(23 INR-21 INR)	(21 INR-18 INR)	(18 INR-14 INR)
Market-16	Time	(5:00-7:00)	(7:00-11:00)	
	Availability	(39 unit-46 unit)	(46 unit-32 unit)	
	Price	(22 INR-19 INR)	(19 INR-14 INR)	
Market-16	Time	(17:00-19:00)	(19:00-22:00)	
	Availability	(28 unit-35 unit)	(35 unit-24 unit)	
	Price	(21 INR-18 INR)	(18 INR-14 INR)	
Market-17	Time	(5:00-7:00)	(7:00-11:00)	(11:00-13:00)
	Availability	(34 unit-45 unit)	(45 unit-23 unit)	(23 unit-17 unit)
	Price	(24 INR-21 INR)	(21 INR-19 INR)	(19 INR-16 INR)

(continued on next page)

Table B.2 (continued).

Market	Parameters	Time slot 1	Time slot 2	Time slot 3
Market-18	Time	(11:00-13:00)	(13:00-17:00)	(17:00-20:00)
	Availability	(36 unit-44 unit)	(44 unit-31 unit)	(31 unit-24 unit)
	Price	(25 INR-23 INR)	(23 INR-20 INR)	(20 INR-16 INR)
Market-19	Time	(5:00-7:00)	(7:00-11:00)	
	Availability	(41 unit-50 unit)	(50 unit-32 unit)	
	Price	(24 INR-22 INR)	(22 INR-17 INR)	

References

- Albalate, D., & Gragera, A. (2020). The impact of curbside parking regulations on car ownership. *Regional Science and Urban Economics*, 81, Article 103518.
- Ali, F., Islam, S. R., Kwak, D., Khan, P., Ullah, N., Yoo, S.-j., et al. (2018). Type-2 fuzzy ontology-aided recommendation systems for IoT-based healthcare. *Computer Communications*, 119, 138–155.
- Angelelli, E., Gendreau, M., Mansini, R., & Vindigni, M. (2017). The traveling purchaser problem with time-dependent quantities. *Computers & Operations Research*, 82, 15–26.
- Arram, A., & Ayob, M. (2019). A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems. *Computers & Industrial Engineering*, 133, 267–274.
- Batista-Galván, M., Riera-Ledesma, J., & Salazar-González, J. J. (2013). The traveling purchaser problem, with multiple stacks and deliveries: A branch-and-cut approach. *Computers & Operations Research*, 40(8), 2103–2115.
- Bernardino, R., & Paisas, A. (2018). Metaheuristics based on decision hierarchies for the traveling purchaser problem. *International Transactions in Operational Research*, 25(4), 1269–1295.
- Bianchessi, N., Irnich, S., & Tilk, C. (2021). A branch-price-and-cut algorithm for the capacitated multiple vehicle traveling purchaser problem with unitary demand. *Discrete Applied Mathematics*, 288, 152–170.
- Bianchessi, N., Mansini, R., & Speranza, M. G. (2014). The distance constrained multiple vehicle traveling purchaser problem. *European Journal of Operational Research*, 235(1), 73–87.
- Castillo, O., Melin, P., Kacprzyk, J., & Pedrycz, W. (2007). Type-2 fuzzy logic: Theory and applications. In *2007 IEEE international conference on granular computing* (p. 145). IEEE.
- Castro, J. R., Castillo, O., & Martinez, L. G. (2007). Interval type-2 fuzzy logic toolbox. *Engineering Letters*, 15(1), 89–98.
- Cheng, G., Wang, C., & Xu, C. (2020). A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing. *Multimedia Tools and Applications*, 79(39), 29243–29263.
- Choi, M. J., & Lee, S. H. (2011). The multiple traveling purchaser problem for maximizing system's reliability with budget constraints. *Expert Systems with Applications*, 38(8), 9848–9853.
- Cruz-Piris, L., Marsa-Maestre, I., & Lopez-Carmona, M. A. (2019). A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem. *IEEE Access*, 7, 111968–111981.
- Cuellar-Usaquén, D., Gomez, C., & Álvarez-Martínez, D. (2021). A GRASP/Path-Relinking algorithm for the traveling purchaser problem. *International Transactions in Operational Research*.
- Cuellar-Usaquén, D., Gomez, C., & Álvarez-Martínez, D. (2023). A GRASP/Path-Relinking algorithm for the traveling purchaser problem. *International Transactions in Operational Research*, 30(2), 831–857.
- Das, M., Roy, A., Maity, S., Kar, S., & Sengupta, S. (2021). Solving fuzzy dynamic ship routing and scheduling problem through new genetic algorithm. *Decision Making: Applications in Management and Engineering*.
- El Khadiri, M., Yeh, W.-C., & Cancela, H. (2023). An efficient factoring algorithm for the quickest path multi-state flow network reliability problem. *Computers & Industrial Engineering*, Article 109221.
- Gaikwad, T. D. (2020). Vehicle velocity prediction using artificial neural networks and effect of real-world signals on prediction window.
- Ha, V.-P., Dao, T.-K., Pham, N.-Y., & Le, M.-H. (2021). A variable-length chromosome genetic algorithm for time-based sensor network schedule optimization. *Sensors*, 21(12), 3990.
- Han, K.-H., & Kim, J.-H. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, 6(6), 580–593.
- Jiang, T., Lu, S., Ren, M., Cheng, H., & Liu, X. (2023). Modified benders decomposition and metaheuristics for multi-machine parallel-batch scheduling and resource allocation under deterioration effect. *Computers & Industrial Engineering*, Article 108977.
- Kang, C.-N., Kung, L.-C., Chiang, P.-H., & Yu, J.-Y. (2023). A service facility location problem considering customer preference and facility capacity. *Computers & Industrial Engineering*, 177, Article 109070.
- Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, 80(5), 8091–8126.
- Keshavarzi, A., Haghigat, A. T., & Bohlouli, M. (2021). Clustering of large scale QoS time series data in federated clouds using improved variable chromosome length genetic algorithm (COGA). *Expert Systems with Applications*, 164, Article 113840.
- Khodadadian, M., Divsalar, A., Verbeek, C., Gunawan, A., & Vansteenwegen, P. (2022). Time dependent orienteering problem with time windows and service time dependent profits. *Computers & Operations Research*, 143, Article 105794.
- Krömer, P., & Uher, V. (2022). Optimization of real-world supply routes by nature-inspired metaheuristics. In *2022 IEEE congress on evolutionary computation* (pp. 1–10). IEEE.
- Kucukoglu, I. (2022). The traveling purchaser problem with fast service option. *Computers & Operations Research*, Article 105700.
- Li, B., Hu, P., Zhu, N., Lei, F., & Xing, L. (2019). Performance analysis and optimization of a CCHP-GSHP coupling system based on quantum genetic algorithm. *Sustainable Cities and Society*, 46, Article 101408.
- Liu, K., Asher, Z., Gong, X., Huang, M., & Kolmanovsky, I. (2019). *Vehicle velocity prediction and energy management strategy part 1: Deterministic and stochastic vehicle velocity prediction using machine learning: Technical report*, SAE Technical Paper.
- Maity, S., Roy, A., & Maiti, M. (2015). A modified genetic algorithm for solving uncertain constrained solid travelling salesman problems. *Computers & Industrial Engineering*, 83, 273–296.
- Manerba, D., & Mansini, R. (2012). The capacitated traveling purchaser problem with total quantity discount. In *Proceedings of Odysseus 2012 conference* (p. 42).
- Manerba, D., & Mansini, R. (2014). An effective metaheuristic for the capacitated total quantity discount problem. *Computers & Operations Research*, 41, 1–11.
- Manerba, D., & Mansini, R. (2015). A branch-and-cut algorithm for the multi-vehicle traveling purchaser problem with pairwise incompatibility constraints. *Networks*, 65(2), 139–154.
- Manerba, D., Mansini, R., & Riera-Ledesma, J. (2017). The traveling purchaser problem and its variants. *European Journal of Operational Research*, 259(1), 1–18.
- Mendel, J. M. (2017). Uncertain rule-based fuzzy systems. *Introduction and New Directions*, 684.
- Mor, B., Shabtay, D., & Yedidson, L. (2021). Heuristic algorithms for solving a set of NP-hard single-machine scheduling problems with resource-dependent processing times. *Computers & Industrial Engineering*, 153, Article 107024.
- Mousavi, S., Afghah, F., Ashdown, J. D., & Turck, K. (2019). Use of a quantum genetic algorithm for coalition formation in large-scale UAV networks. *Ad Hoc Networks*, 87, 26–36.
- Obulesu, V., Kumar, P. P., & Babu, G. N. R. (2022). An empirical study on rural marketing opportunities and threads in India. *Mathematical Statistician and Engineering Applications*, 71(4), 8403–8409.
- Palomo-Martínez, P. J., & Salazar-Aguilar, M. A. (2019). The bi-objective traveling purchaser problem with deliveries. *European Journal of Operational Research*, 273(2), 608–622.
- Papalitsas, C., Kastampolidou, K., & Andronikos, T. (2021). Nature and quantum-inspired procedures—a short literature review. *GeNeDis 2020*, 129–133.
- Park, K.-m., Shin, S.-h., Shin, D., Chi, S.-d., et al. (2019). Design of warship simulation using variable-chromosome genetic algorithm. *Applied Sciences*, 9(19), 4131.
- Pradhan, K., Basu, S., Thakur, K., Maity, S., & Maiti, M. (2020). Imprecise modified solid green traveling purchaser problem for substitute items using quantum-inspired genetic algorithm. *Computers & Industrial Engineering*, 147, Article 106578.
- Ramesh, T. (1981). Traveling purchaser problem. *Opsearch*, 18(1–3), 78–91.
- Roy, A., Gao, R., Jia, L., Maity, S., & Kar, S. (2020). A noble genetic algorithm to solve a solid green traveling purchaser problem with uncertain cost parameters. *American Journal of Mathematical and Management Sciences*, 40(1), 17–31.
- Roy, A., Manna, A., Kim, J., & Moon, I. (2022). IoT-based smart bin allocation and vehicle routing in solid waste management: A case study in South Korea. *Computers & Industrial Engineering*, 171, Article 108457.
- Tabrizi, A. M., Vahdani, B., Etebari, F., & Amiri, M. (2023). A three-stage model for clustering, storage, and joint online order batching and picker routing problems: Heuristic algorithms. *Computers & Industrial Engineering*, Article 109180.
- Talbi, H., Draa, A., & Batouche, M. (2004). A new quantum-inspired genetic algorithm for solving the travelling salesman problem. In *Industrial technology, 2004. IEEE ICIT'04. 2004 IEEE international conference on* vol. 3 (pp. 1192–1197). IEEE.
- Tayyaba, S., Ashraf, M. W., Alquthami, T., Ahmad, Z., & Manzoor, S. (2020). Fuzzy-based approach using IoT devices for smart home to assist blind people for navigation. *Sensors*, 20(13), 3674.
- Ullah, I., Youn, H. Y., & Han, Y.-H. (2021). Integration of type-2 fuzzy logic and Dempster-Shafer theory for accurate inference of IoT-based health-care system. *Future Generation Computer Systems*, 124, 369–380.

- Wan, J., He, B., Wang, D., Yan, T., & Shen, Y. (2019). Fractional-order PID motion control for AUV using cloud-model-based quantum genetic algorithm. *IEEE Access*, 7, 124828–124843.
- Wang, H., & Alidaee, B. (2023). A new hybrid-heuristic for large-scale combinatorial optimization: A case of quadratic assignment problem. *Computers & Industrial Engineering*, Article 109220.
- Wang, Y., & Wei, C. (2021). Design optimization of office building envelope based on quantum genetic algorithm for energy conservation. *Journal of Building Engineering*, 35, Article 102048.
- Wang, J., Zhang, M., Ersoy, O. K., Sun, K., & Bi, Y. (2019). An improved real-coded genetic algorithm using the heuristical normal distribution and direction-based crossover. *Computational Intelligence and Neuroscience*, 2019.
- Xiao, X., Yan, M., Basodi, S., Ji, C., & Pan, Y. (2020). Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. arXiv preprint [arXiv:2006.12703](https://arxiv.org/abs/2006.12703).
- Yang, Y., Chen, J., Ye, J., Chen, J., & Luo, Y. (2023). Joint optimization of facility layout and spatially differential parking pricing for parking lots. *Transportation Research Record*, Article 03611981221145139.
- Zhu, X., Xiong, J., & Liang, Q. (2018). Fault diagnosis of rotation machinery based on support vector machine optimized by quantum genetic algorithm. *IEEE Access*, 6, 33583–33588.