IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 21 March 2024, accepted 16 May 2024, date of publication 23 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3404861

RESEARCH ARTICLE

A Pattern Recognition Lexi-Search Approach to the Variant Traveling Purchaser Problem

VENKATA PRATHAP DANAVULAPADU AND PURUSOTHAM SINGAMSETTY
Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India
Corresponding author: Purusotham Singamsetty (drpurusotham.or@gmail.com)

ABSTRACT The traveling purchaser problem (TPP) is a generalization of the well-known traveling salesman problem. The purchaser is aware of the travel cost between a pair of markets and between a depot and markets, the purchase cost and availability of a product at each market, and the demand for the products. The purchaser has the choice of selecting a subset of markets, where the products and how much quantity to be purchased. The objective is to determine an optimal tour for a purchaser that begins and ends at the depot to purchase a set of products from a subset of markets such that the sum of travel and purchase costs is least within the threshold values on the maximum number of markets visited and the number of products purchased at any one market. The problem involves three interesting plans, such as the selection plan, the routing plan, and the purchasing plan. This problem is frequently faced by the shoppers and it has several applications in different domains including production scheduling, transportation, network design, machine scheduling, manufacturing, etc. This paper presents an extended version of TPP along with its mathematical modeling using an integer programming and a deterministic pattern recognition lexi-search algorithm to solve optimally. To test the efficiency of the algorithm, the experiments are carried out on distinct large size benchmark data sets. The extensive comparative computational results show that the proposed algorithm is capable of finding improved solutions on several benchmark data sets reported for capacitated and uncapacitated instances.

INDEX TERMS Traveling salesman problem, traveling purchaser problem, integer programming, pattern recognition Lexi-search algorithm.

I. INTRODUCTION

The traveling salesman problem (TSP) is a well-known combinatorial NP-hard optimization problem. In TSP, a set of cities and the distance between every pair of cities are given. The salesman starts at the depot point and wants to visit each city exactly once and returns to the depot point. The objective is to find a closed tour for the shortest possible distance that connects the given cities. The dashed line in Figure (1) refers to one of the closed tours of TSP with 5 cities. One of the extensions of this problem is the traveling purchaser problem (TPP), firstly coined by Ramesh [1], the problem involves a set of markets and a set of products. Each product's cost and the cost of travel between the different marketplaces are known. The purchaser has to find a tour with a subset of markets such that the total cost of travel and cost of purchasing all the products is minimum. Moreover, he implicitly assumed that if a product is available at a given market, then its available quantity is sufficient to meet the product's demand at that market, referred to as the un-capacitated version and at least one product should be purchased from each of the markets in the subset. In practice, these assumptions may lead to a higher cost. Although a (few) product(s) is (are) available in different markets with varied capacities the purchaser cannot make his complete purchase with lower cost to meet the requirement as partial purchase is not permitted in the model. The dashed line in Figure (2) gives a tour with a subset of markets for the purchaser. To find an optimal solution to this problem a lexicographic search (LS) algorithm was proposed. While this algorithm is exact in general, the computational details are limited to small instances of sizes up to 12 markets with 10 products only.

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

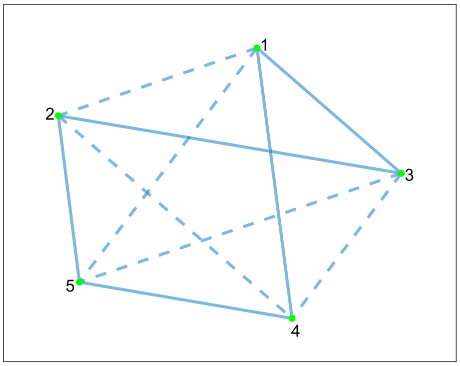

FIGURE 1. A tour of TSP with 5 markets.

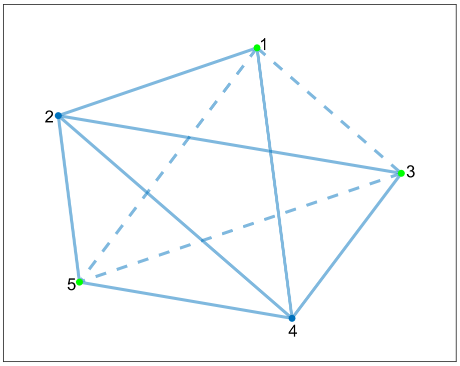

FIGURE 2. A tour of TPP with a subset of markets.

In addition to the lexicographic method of Ramesh [1], two more exact algorithms are available. Singh and Oudheusden [2] have developed a branch and bound (BnB) algorithm for the TPP. This algorithm involves two phases; one is the selection phase and the other one is the search phase. In the selection phase, a subset of markets will be selected and in the search phase determined an optimal solution with the selected markets using effective branching and bounding strategies. The experiments carried out are limited to the instances of sizes up to 25 markets with 25 products. Later on, Laporte et al. [3] developed a branch and cut (BnC) algorithm for the TPP which is capable of solving instances of sizes up to 200 markets with 200 products.

Pearn and Chien [4] addressed two interesting applications of TPP, the first one is manufacturing factories want to minimize the cost of purchasing different raw materials required from the various warehouses located in different areas with varied costs, in this case, the purchaser has to select a subset of the warehouses to be visited, make a tour plan with the selected warehouses to minimize the sum of travel and purchase costs. The second one is in the context of scheduling of a set of jobs over some machines with different set-up and job processing costs in which the total cost for completing the jobs has to be minimized. To provide improved solutions for the TPP, several modifications includes 'Next bloc and Next-neighbor search', the 'Parameter-selection and Tie-selection', 'Adjusted cheapest and the nearest cheapest' and 'Random order and sequence order' suggested to the existing four algorithms Lexicographic search [1], Generalized-savings heuristic (GSH) [5], Tour-reduction heuristic (TRH) [6] and Commodity-adding heuristic (CAH) respectively. The computational results show that the random-order commodity adding algorithm was obtained better solutions than the other three algorithms.

Boctor et al. [7] deals with two versions of TPP. They are un-capacitated TPP (UTPP) and capacitated TPP (CTPP). In UTPP, the quantum of availability of a product at each market is larger than or equal to its demand and in CTPP the supply may be lower than the demand. They have designed distinct perturbation heuristics by integrating with several procedures such as market drop, market add, market exchange, etc. for both the versions of TPP and tested on the instances with 50–350 markets and 50–200 products. The extensive comparative computational results reveal that the perturbation heuristics produced better solutions. Teeninga and Volgenant [8] described improvement sub-procedures that can be incorporated into the existing heuristics GSH, CAH, and TRH. The Computational results show the effectiveness of these sub-procedures on the test instances of sizes up to 200 markets with 200 products.

Riera-Ledesma and Salazar-González [9] introduced a new heuristic approach based on a local search scheme, exploring a new neighborhood structure for the TPP and experimented on instances involving up to 350 markets with 200 products within a reasonable amount of time. Another variant of TPP is the asymmetric traveling purchaser problem (ATPP) solved by Riera-Ledesma and Salazar-González [10]. A modified branch and cut algorithm was proposed to solve ATPP and tested the instances of size up to 200 markets with 200 products within acceptable amount of time. Bontoux and Feillet [11] solved the TPP using the ant colony optimization (ACO) technique combined with a local search procedure exploring a new neighborhood structure. This technique is tested using a set of benchmark instances ranging from 50 to 350 markets with 50 to 200 products. The experiments carried out on these instances are compared with the results of Riera-Ledesma and Salazar-González [9] which give improved solutions on certain test instances. Later, Riera-Ledesma and Salazar González [12] solved a school bus routing problem using a multivehicle traveling purchaser problem (MVTPP). The problem has been addressed by using a flow-based formulation that combines continuous and binary variables through the coupling constraints. A branch and cut algorithm was developed and that has been tested on different instances of sizes up to 125 bus stops with 125 student locations.

Goldbarg et al. [13] presented a transgenetic algorithm (TA) for the TPP. It is inspired by two significant evolutionary driving forces: one is horizontal gene transfer and another one is endosymbiosis. Computational experiments indicate that the TA method is very effective for the investigated problems with 17 and 19 new best solutions from 440 and 89 instances reported for CTPP and UTPP respectively.

Gouveia et al. [14] studied a variant of TPP with additional side constraints, where the purchaser should not visit more than a predefined number of markets, the number of items purchased at each market cannot exceed the given threshold value and where only one copy per item needs to be bought. The problem is nicely described with integer linear programming and solved instances up to 300 markets with 200 products with the help of dynamic programming and Lagrangian relaxation techniques.

Several variants of TPP have evolved in the last two decades. Mansini and Tocchella [15], considered the TPP with budget constraint and developed two heuristics namely, efficient enhanced local search and variable neighborhood search. Both the heuristics are implemented on a large set of benchmark problems with different control parameters on budget constraints for both the UTPP and CTPP and reported the detailed experimental results. Angelelli et al. [16], [17] addressed a dynamic TPP, where the availability of a product at a market decreases over the time and used different greedy algorithms to solve it. TPP with multiple stacks and deliveries that is similar to the one-to-one pickup and delivery vehicle routing problem, introduced by Batista-Galván et al. [18]. Sumathi et al. [19], studied a variant of TPP in which the purchaser aims to find a tour with minimum time such that the total purchase cost of all products does not exceed the predefined budget limit and developed the lexi-search algorithm to obtain an optimal solution. Kucukoglu [20], presents a new variant of TPP with a fast service option, in which the tour must be closed within a period that considers the purchaser's traveling and purchasing times. An adaptive large neighborhood search algorithm (ALNS) is presented and tested on the performance of this algorithm using different-sized benchmark problems of TPP and computational results are reported. A few more interesting variants of TPP are listed here: TPP with environmental transportation cost Kang et al. [21], Sustainable TPP Cheaitou et al., [22], Dynamics of rebound effects in TPP Caballero et al. [23], Multi-depot TPP under shared resources in logistics Jesri et al. [24].

More recently, a TPP with transportation time limit is studied by Kucukoglu et al. [25] in which the purchaser wants to collect a set of perishable products, which may deteriorate due to delay in travel time, from the different capacitated markets. For this he has to determine a route plan and procurement plan with minimum cost subject to satisfying the transportation time limits. A heuristic based Tabu search algorithm is used to obtain the near optimal solutions in a reasonable amount of computational time. A multi-vehicle clustered TPP Roy et al. [26]., is an extension of TPP, where the set of markets be divided into distinct clusters with the help of the k-means algorithm. In this problem, there are two plans for the purchaser in procuring the products. In the first case, the purchaser has to procure the products by visiting the markets cluster-wise, returns to the depot, and the entire procurement will be carried along the same path of the purchaser by a different vehicle. In the second case, the products that are purchased in each cluster will be transported through vehicles from the center of the cluster to the depot directly. A variable-length chromosome genetic algorithm (VLC-GA) is proposed to determine the optimal cluster paths and then a local heuristic is used to connect the clusters which minimizes the total cost involved in the system.

The researchers mostly focused on different TPP variants as it contains the selection plan refers to selecting a subset of marketplaces with a depot point, the routing plan refers to the ordering of markets selected to be visited to minimize the cost of travel, and the purchase plan refers to how much quantity to be purchased at each market place together with the purchase cost. Also, several practical applications of these plans involved in transportation, manufacturing industry, telecommunication network design, and machine scheduling have been introduced in the literature by Manerba et al. [27]. Ong [6] described an interesting application of TPP in a scheduling context that looks for an optimal sequence of n-jobs on a machine that has m-states and is solved using TRH. However, the performance of TRH greatly depends on the selection of the initial subset of markets. Another application of TPP in production scheduling can be seen in Buzacott and Dutta [28]. Table–1 provides a summary of the existing literature on the TPP versus the present study. It is observed that there is little attention was given to the studies on TPP with additional side constraints due to its high complexity in finding the solution. The present TPP variant studies both UTPP and CTPP cases by considering the additional side constraints η and δ in the model, whereas β implicitly appears in the objective function. Therefore, the present form of study is an extended version of the most of TPP variants listed. In addition, this contributes a deterministic pattern recognition lexi search (PRLS) algorithm, which eliminates all the infeasible patterns with the help of effective bounding strategies and provides the optimal solution.

The remaining paper is organized as follows: Section II provides a detailed problem description of the present variant of TPP and its mathematical representation of integer programming. Section III presents the discussion on the terminologies used in lexi search and the proposed Pattern recognition lexi search algorithm. Section IV includes the numerical example of TPP and the systematic search mechanism of PRLS to find an optimal solution of TPP. The extensive computational results are reported in section V and followed by the concluding remarks of TPP.

II. PROBLEM DESCRIPTION AND MATHEMATICAL FORMULATION

Let $G = (V, E, W)$ be a general directed /undirected weighted complete graph, where $V = \{1, 2, \ldots, m\}$ be a set of vertices, represented as markets with a depot point (say 1), $E = \{e_{ij}/i, j \in Vi \neq j\}$ be an edge set contains an edge e_{ij} that connects a pair of markets i and j. W be the collection of weight factor on each edge e_{ij} defines the travel cost C_{ij} from market i to market j. In a graph G, each pair (i, j) , i, j ∈ V of markets is connected with an edge

TABLE 1. A brief summary on the most relevant works on TPP to the present work.

Author(s)	Objective function	Version		Additional constraints			Methodology	
		UTPP	CTPP	η	δ	β	Method	Type
proposed	Min. (TC+PC)	Y	Y	Y	Y	N	PRLS	Exact
[1]	Min. (TC+PC)	Y	N	N	N	N	LS	Exact
[2]	Min. (TC+PC)	Y	N	N	N	N	BnB	Exact
[3]	Min. (TC+PC)	Y	Y	N	N	N	BnC	Exact
[7]	Min. (TC+PC)	Y	Y	N	N	N	Local search	Heuristic
[8]	Min.(TC+PC)	N	Y	N	N	N	GSH/ TRH/ CAH	Heuristic
[9]	Min. (TC+PC)	Y	Y	N	N	N	Local search	Heuristic
[10]	Min. (TC+PC)	N	Y	N	N	N	BnC	Exact
[11]	Min. (TC+PC)	Y	N	N	N	N	ACO	Heuristic
[13]	Min. (TC+PC)	Y	Y	N	N	N	TA	Heuristic
[14]	Min. (TC+PC)	Y	N	Y	Y	N	DP	Exact
[15]	Min.(TC)	Y	Y	N	N	Y	VNS	Heuristic
[19]	Min.(Travel time)	N	Y	N	N	Y	PRLS	Exact
[20]	Min. (TC+PC)	N	Y	N	N	N	ALNS	Heuristic

TC: Total travel cost; PC: Total purchase cost; η: maximum number of markets to be visited; δ: maximum number of products purchased at any market; β: budget threshold value; Y: Yes; N: No

$e_{ij} \in E$ uniquely then it is called a complete graph. A weighted complete graph is a graph in which each edge is assigned with a weight factor. If $C_{ij} = C_{ji}, \forall i, j \in V$, then the graph is called undirected weighted complete graph otherwise the graph refers to directed weighted complete graph. A TPP is referred to symmetric or asymmetric according to the graph G is undirected or directed. In addition, at each market a set of products $K = \{1, 2, \ldots, n\}$ be available with varying quantity, denoted by A_{kj}. The unit cost of product k at market j is denoted by P_{kj} and it can be varied in different markets and D_k represents the known demand for each product k. At each market j, if $A_{kj} \geq D_k, \forall k \in K$ the problem refers to UTPP, otherwise it refers to CTPP. In addition, two more additional constraints are considered in the model. One is maximum allowable markets $\eta \leq m$ to be visited to purchase the required products and the second one is the maximum permitted number of products $\delta \leq n$ to be purchased from any one market. The purchaser starts and ends at the depot point, visits a subset of markets S of V exactly once, and purchases his complete demand at one or more markets subject to satisfying the additional constraints, before returning to the depot. Clearly, the cardinality of S is $|S| = \gamma, 2 \leq \gamma \leq \eta$ and S must contain the depot. The quantum of units of product k purchased at market $j \in S$ is denoted by Z_{kj}, assumes a non-negative integral value. The objective of the purchaser is to find an optimum tour with a subset of markets that provides the minimum cost of the sum of travel and purchase costs together. Thus, the model involves three important plans. The first one is *selection plan* refers to which markets are supposed to be visited by the purchaser, the second one is *the routing plan* which includes the ordering of the markets and the third one is the *purchasing plan* refers to how much quantity and where to be purchased. If the purchaser visits market j from the market i then the binary variable X_{ij} assumes a value 1 otherwise 0. If the purchaser wishes to purchase product k upon visiting the market j, then $Y_{kj} = 1$, otherwise $Y_{kj} = 0$.

The following assumptions are used to model the TPP

- The travel cost between the markets, demand of products, availability and cost of products at each market are predefined.
- The purchaser can select a subset S of markets along with the depot point arbitrarily.
- The purchaser should start from a depot, visit each market in the set S exactly once and return to the depot.
- The purchaser will not be allowed to take any illegal tours or unwanted tours.
- The demand of k^{th} product must be lower than the total availability of k^{th} product at all markets to set a feasible solution.
- The complete demand of a product can be purchased in a single market in the case of UTPP, while a product can be purchased in multiple markets to meet the actual demand in the case of CTPP.
- The purchaser can visit a market without any purchase of products.

The mathematical formulation using the integer linear programming of the proposed TPP is given in equations (1)–(12). The variables X_{ij}, Y_{kj} and Z_{kj} appeared in the model are called the decision variables.

$$Minimize \sum_{i \in V} \sum_{j \in V} C_{ij} X_{ij} + \sum_{k \in K} \sum_{j \in S} P_{kj} Z_{kj} \quad (1)$$

Subject to the constraints

$$\left.\begin{aligned}\sum_{j=1}^{m} X_{1j} = 1\\ \sum_{i=1}^{m} X_{i1} = 1\end{aligned}\right\} \tag{2}$$

$$\left.\begin{aligned}\textstyle\sum_{j\in S} X_{ij} = 1, i \in S\\ \textstyle\sum_{i\in S} X_{ij} = 1, j \in S\end{aligned}\right\} \tag{3}$$

$$\sum_{i\in S} X_{ip} - \sum_{j\in S} X_{pj} = 0, p \in S \tag{4}$$

$$\sum_{i\in S}\sum_{j\in S} X_{ij} \le \gamma - 1 \tag{5}$$

$$D_k \le \sum_{j=1}^{m} A_{kj}, k \in K \tag{6}$$

$$Z_{kj}Y_{kj} \le A_{kj}, j \in S, k \in K \tag{7}$$

$$\sum_{j\in S} Y_{kj}Z_{kj} = D_k, k \in K \tag{8}$$

$$\sum_{j\in S} Y_{kj} \ge 1, k \in K \tag{9}$$

$$\sum_{k=1}^{n} Y_{kj} \le \delta \sum_{i\in S} X_{ij}, j \in S \tag{10}$$

$$X_{ij} \in \{0, 1\}, i, j \in V \tag{11}$$

$$Y_{kj} \in \{0, 1\}, j \in S, k \in K \tag{12}$$

The objective of the purchaser is to minimize the sum of travel and purchase costs, which is given in (1). Constraint (2) specifies that the purchaser starts from the depot and returns to the depot exactly once. Constraint (3) denotes that the purchaser enters the market j and departs the market i in S exactly once. However, the constraints (2) and (3) together do not guarantee the continuity of a tour of the purchaser. Constraint (4) enforces to maintain the tour's continuity but it does not prevent the formation of partial (or) illegal tours with the markets in S. For example, let $S = \{1, 2, 3, 4, 5\}$. Case 1, the edges corresponds to the pair of markets $(1, 3)$, $(4, 5)$, $(5, 2)$, $(2, 4)$, $(3, 1)$ forms multiple tours, satisfy the constraints (2)–(4), but this not a feasible trip for the purchaser. Case 2, the edges corresponds to the pair of markets $(1, 3)$, $(4, 5)$, $(5, 1)$, $(3, 4)$ forms a continuous trip, yet it is not feasible as it does not include the market 2, referred as an illegal tour. To identify the occurrence of such multiple tours or illegal tours Constraint (5) is enforced. If Constraint (5) satisfied then it indicates the occurrence of such illegal tours which are to be discarded. To ensure the existence of a feasible solution constraint (6) is added i.e., if the demand of a product k is supposed to be lower than or equal to the sum of its availabilities at all the markets, then there will be at least one solution, otherwise, the problem has no feasible solution. Constraint (7) tells us that the quantum of purchase Z_{kj} of a product k at market j do not exceed its availabilities at the market j. Constraint (8) states that the total quantum

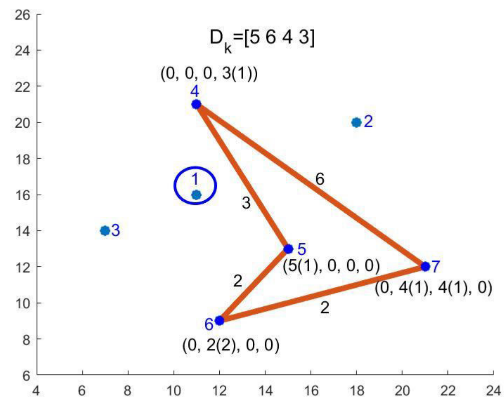

FIGURE 3. An illegal tour without a depot.

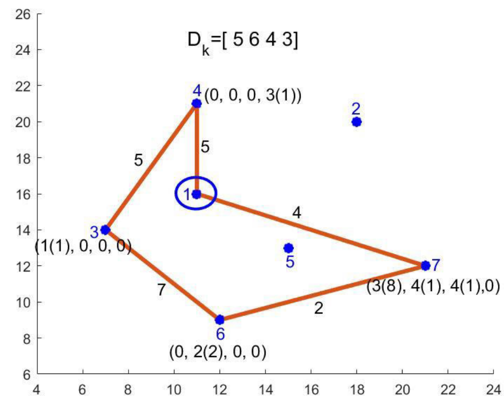

FIGURE 4. An illegal tour with a depot.

of purchase of each product k must be equal to its actual demand. Constraint (9) says that the purchaser can purchase a product in one or more markets to fulfill the demand. Constraint (10) is imposed to control the maximum number of products purchased at any one market. Finally, the binary variables X_{ij} and Y_{kj} which are used in the model are given in (11) and (12). For better understanding of the mathematical formulation, a detailed discussion on the formation infeasible or illegal tours, feasible tours and an optimal tour of the purchaser shown graphically in Figures (3)–(6).

III. LEXI SEARCH ALGORITHM

The lexi-search algorithm has been used successfully to a numerous combinatorial optimization problems. In general, the lexi-search algorithm examines all possible solutions in a hierarchy meaning that it performs the search for an optimal solution systematically that is similar to the search for meaning of a word in a dictionary. Firstly, an alphabet table will be constructed by sorting the entries in the travel cost matrix C. This alphabet table contains a collection of letters representing all possible ordered sequences of entities as travel cost, cumulative cost, row and column indices of respective cost value as markets. A partial word is the combination of letters that appear in the alphabet table. A partial word which gives

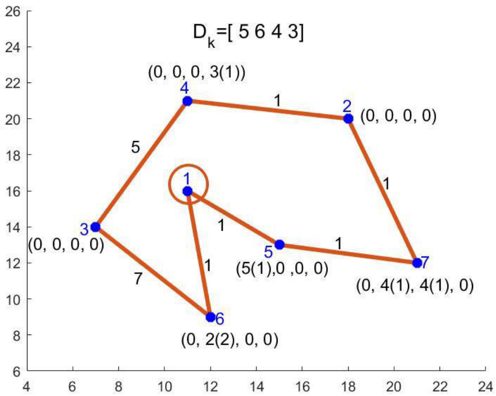

FIGURE 5. Initial feasible tour.

a tour and satisfies all the feasibility restrictions is called a feasible word, otherwise it is infeasible. In this process, finding which combination of letters from the alphabet table gives the best solution is the challenge. In general, the lexi-search algorithm uses less memory, because of the existing lexicographic order of partial words. The proposed algorithm carefully examines all the partial words and identifies a sequence of markets which will form a feasible tour for the purchaser. When the process of checking the feasibility of a partial word becomes difficult, although calculating the lower bound is easy, a pattern recognition technique (Murthy [29]) can be integrated with the lexi-search algorithm to get the optimal solution. The performance of the pattern recognition lexi-search algorithm depends on the choice of an appropriate alphabet table. In this case, two contradictory features from the search list should be considered:

i. The process of verifying the feasibility of a partial word is simple, however calculating a lower bound is difficult and
ii. The computation of the lower bound is simple, even though the feasibility analysis is difficult.

In this method, the algorithm's effectiveness is increased by calculating bounds at first and then checking for the feasibility of a partial word whose value is less than the upper bound. Although, the general structure of the proposed algorithm appears similar to Ramesh [1] but it is significantly differs in the construction of the alphabet table and the bounding strategies used, which in turn helps to produce the better solutions for higher dimension instances too.

A. PATTERN

A pattern is an indicator of a two-dimensional array X which contains the binary entries and is associated with the partial word. If the pattern X is feasible, then X is said to be a feasible pattern, otherwise X is an infeasible pattern. The objective function value concerning the feasible pattern X denoted by F(X), it will be computed using the equation (13) and it gives the sum of travel cost (TC) and purchase cost (PC).

$$F\left(X\right)=TC+PC=\sum_{i\in V}\sum_{j\in V}C_{ij}X_{ij}+\sum_{k\in K}\sum_{j\in S}P_{kj}Z_{kj}\quad(13)$$

B. ALPHABET TABLE

The elements in the travel cost matrix $C = \left[C_{ij}\right]$ be arranged in increasing order and indexed from 1 to m^2 to generate an alphabet table. Table–4, shows the construction of an alphabet table for the numerical example given in section IV. The columns in Table–4 are labeled as I, C, CC, r and c, respectively denote the index of travel cost of the purchaser, cumulative cost, row and column indices. Let $I = \{1, 2, 3, \ldots, m^2\}$ be the collection of $m \times m$ ordered indices and each of this index is called a letter. A letter $a_i = (C, CC, r, c) \in I$ represents an ordered sequence of entities associated with the travel cost matrix. Let $\alpha_p = (a_{1,} a_{2,} \ldots, a_p)$ be an ordered sequence of p letters from I represent a partial word of length $p(\leq \gamma)$ and a_p is called the leader of α_p. A letter $a_i, i = 1, 2, \ldots, p$ can occupy any position in α_p. For uniqueness, the letters in α_p are arranged in increasing order, that is $a_i < a_{i+1,} i = 1, 2, \ldots, p-1$. The pattern X is associated with α_p. If X is a feasible pattern then α_p is feasible otherwise infeasible. Enumerating all the feasible patterns in the entire search space is too expensive when the size of the instance is large. Therefore, the algorithm implicitly identifies the feasible patterns within the limited search space with the help of the effective bound settings.

C. BOUND SETTINGS

The value of a partial word $V(\alpha_p)$ is given by $V\left(\alpha_p\right) = V\left(\alpha_{p-1}\right) + C\left(\alpha_p\right)$, with $V(\alpha_0) = 0$. Let S be the subset of markets, associated with the letters in α_p and $|S| = \gamma$. Let LB and UB respectively, denotes the lower and upper bounds of α_p. To accelerate the search, the LB of a partial word α_p is determined using the equation (15). Set $UB = \infty$ initially or for an effective UB, find a random feasible pattern X, and then set $UB = F(X)$. The algorithm intelligently eliminates all the infeasible patterns and searches for an optimal solution within the range of LB and UB. If there is no further improvement to the initial UB indicates that either the problem has no solution or the initial pattern will be the optimum solution.

$$PC=\sum_{k\in K}\sum_{j\in S}P_{kj}Z_{kj}\quad(14)$$

$$LB\left(\alpha_p\right)=V\left(\alpha_p\right)+CC\left(a_p+\gamma-p\right)-CC\left(\alpha_p\right)+PC\quad(15)$$

where PC be the purchase cost and is calculated using an equation (14). The algorithmic steps involved in finding the PC are given in the subsection III-D as well.

D. CALCULATION OF AN INITIAL PURCHASE COST

Step 1: Assign $TA = \left[A_{kj}\right]$ and $TD = [D_k]$, $k \in K, j \in S$
$Z = [Z_{kj}] \leftarrow$ null matrix of order n $\times m$
Let us start with $k = 1$ and $PC = 0$ go to step 2.
Step 2: Select the least cost cell in P_{kj} say (k, j) go to step 3.
Step 3: Calculate $Z_{kj} = \min\left(TA_{kj,} TD_k\right)$
$TA_{kj} = TA_{kj,} - Z_{kj}$
$TD_k = TD_k - Z_{kj}$
$PC = PC + P_{kj} * Z_{kj}$

If $TD_k = 0$ go to step 4.
else select the next least cost cell in P_{kj} say (k, l) , $j = l$ repeat step 3.

Step 4: $k = k + 1$, if $k > n$, stop and record the PC, else go to step 2.

E. PROPOSED PATTERN RECOGNITION LEXI-SEARCH ALGORITHM

The step-by-step search procedure of the pattern recognition lexi-search algorithm is described in the sequel as follows:

Step 1: **Initialization**
$m \leftarrow$ number of markets
$n \leftarrow$ number of products
$\eta \leftarrow$ maximal allowed number of markets to be visited
$\delta \leftarrow$ maximal allowed number of products to be purchased at any one market
$C = [C_{ij}] \leftarrow$ travel cost matrix
$D = [D_k] \leftarrow$ Demand of products
$A = [A_{ki}] \leftarrow$ product's availability matrix
$P = [P_{ki}] \leftarrow$ product's cost matrix
$UB = \infty$ (a large value or a random solution)

Step 2: If equation (6) holds then the problem will have at least one solution, go to step3, else there is no solution to the problem, go to step 15.

Step 3: **Alphabet table:**
Construct an alphabet table using the given travel cost matrix C as discussed in Section III-B and go to step 4.

Step 4: Calculate the initial purchase cost as discussed in section III-D go to step 5.

Step 5: (i) The algorithm starts with a partial word $L(\alpha_p) = (a_p) = 1$, where the length of thepartial word α_p is one, that is, $p = 1$ go to step 5(ii).
(ii) If $p = 1$, max $= m^2 - m - 1$ and if $a_p >$ max, go to step 15, else go to step 6. Else max $= m^2 - m - \gamma + p$, go to step 5 (iii).
(iii) If $a_p >$ max, go to step 13, else, go to step 6.

Step 6: **Bound settings**
Determine the lower bound (LB) of a partial word α_p, using the equation (15), go to step 7.

Step 7: If $LB(\alpha_p) < UB$ go to step 8, else go to step 13.

Step 8: **Feasibility checking**
The sequence α_p is a partial word. Read the row and column indices of the letter a_p in α_p. If the pattern X corresponds to α_p satisfy the equation (3), then it is said to be a partial feasible word go to step 9,
else it is infeasible discard the current letter and then continue the search by considering the next immediate letter a_{p+1} in p$^{\text{th}}$ position of α_p, go to step 5(iii).

Step 9: **Concatenation**
Consider the letter a_p to be a member of α_p. If α_p forms a tour or a sub-tour (satisfy equation (4)) then go to step 10.
Else $\alpha_{p+1} = \alpha_p * a_{p+1}$, where $*$ is a concatenation operation, update the markets in the set S that are associated with α_p, go to step 5(ii).

Step 10: If α_p included the home city (satisfies an equation (2)) then it is said to be a partial feasible tour, go to step 11.
Else it is an illegal tour (satisfy an equation (5)) then discard the current letter and then continue the search by considering the next immediate letter, go to step 5(iii).

Step 11: If the pattern X corresponds to a tour of α_p satisfy the constraints (7)-(10) then the pattern X will be a feasible solution. Update the PC along the tour (as discussed in section III-D) and hence find the value of the pattern $F(X)$ using an equation (13), go to step 12.
Else it is an illegal tour then discard the current letter and then continue the search by considering the next immediate letter a_{p+1}, go to step 5(iii).

Step 12: If $F(X) < UB$, then record the improved solution as $UB = F(X)$, go to step 13, else discard the current letter and further continue the search by considering the next immediate letter a_{p+1} in p$^{\text{th}}$ position of α_p, go to step 5(iii).

Step 13: If $p = 1$ go to step 15, else go to step 14.

Step 14: **Perform the backtracking**
Set $p = p - 1$, continue the search until the optimum solution is obtained by considering the next immediate letter a_{p+1} in the p^{th} position of α_p, go to step 5(ii).

Step 15: **Stop the search**
In brief the PRLS contains different steps such as construction of alphabet table, and in search process first calculate the effective bounds for the pattern α_p, next checks for feasibility of α_p, perform the concatenation if it is feasible and record the feasible solution, then carefully use the backtracking operation until the optimal solution is obtained.

IV. NUMERICAL ILLUSTRATION

For better understanding the proposed TPP variant and the concept of the pattern recognition lexi-search algorithm, a suitable numerical example is considered from Goldberg et al. [13]. Let there are $m = 7$ markets, where market 1 is assumed to be a depot. The travel cost of the purchaser between a pair of markets is given in Table–2, in which the cell entries appear to be symmetric and the notion ∞ indicates no direct connectivity between a pair of markets (or) self-loops. The purchaser wants to purchase $n = 4$ products that are available at distinct markets. The demand of a product, the availability and unit cost of each product is given in Table–3. Observe that some products may not be available in certain markets, the quantum of availability and the product's cost differs from market to market. Now, the purchaser starts from depot, travel to different markets to purchase all the four products with the desired demand and then returns to the

TABLE 2. The travel cost matrix (*C*).

	1	2	3	4	5	6	7
1	∞	4	5	5	1	1	4
2	4	∞	2	1	4	9	1
3	5	2	∞	5	9	7	2
4	5	1	5	∞	3	8	6
5	1	4	9	3	∞	2	1
6	1	9	7	8	2	∞	2
7	4	1	2	6	1	2	∞

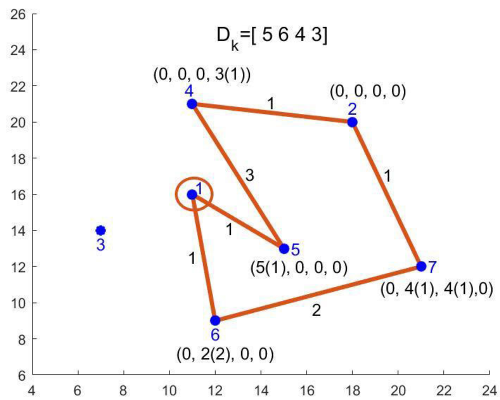

FIGURE 6. Optimal tour.

depot. Initially, set $\eta = 7$ and $\delta = 2$. His objective is to minimize the sum of total travel and purchase costs on a single tour subject to satisfying all the constraints. A solution of TPP gives us a route with a subset of markets which employs the purchase plan and route plan to the purchaser. In order to get the solution of TPP through the proposed modified PRLS first construct an alphabet table as discussed in section III-B and is shown in Table–4.

A. SEARCH TABLE

The systematic search mechanism of the proposed modified PRLS is illustrated in detail in Table–5. The columns 1 through 13 in table–5 respectively represent a serial number, followed by seven columns containing a leading letter a_p in the partial word α_p, next two columns V and LB give the value and lower bound of α_p. The columns labeled r and c state the row and column index of the letter a_p. The last column shows the remarks regarding the acceptance(A) and rejectance (R) of a letter a_p for inclusion in a feasible word α_p. The algorithm starts with the letter $a_1 = 1$, calculate the bounds first and then check the feasibility. In the sequel, the algorithm effectively examines the feasibility of the letters in succession, discards the infeasible letters and inducts the letters into a partial word α_p which are feasible. If α_p be a full length feasible word, then update the value of UB. The ordered sequence pattern $\alpha_p = (1, 3, 7, 8, 9, 32, 35)$ gives the initial feasible solution for the purchaser and is observed in 35th row of Table–5. Further, the algorithm continues the search for the other possible improved or optimal solution with the help of effective backtracking and bounding strategy. In the search, three more improved solutions were obtained and they observed correspond to row 41, row 103 and row 279 of Table–5. The details of the pattern, route plan and purchase plan of these solutions are shown in Table–6. As there is no further improvement to the solution observed in row 279 of Table–5, this solution will become the optimum solution.

B. ILLEGAL, FEASIBLE, AND OPTIMAL SOLUTIONS

The graphical representation of a sub-tour with 4–markets is shown in Figure (3), in which, the markets scattered on a Euclidean plane are indexed from 1 to 7 and the encircled point represents the depot point of the purchaser. The vector D_k, contains the demand of each product. An ordered 4–tuple $\left(Z_{1j}\left(P_{1j}\right), Z_{2j}\left(P_{2j}\right), Z_{3j}\left(P_{3j}\right), Z_{4j}\left(P_{4j}\right)\right)$ at market j along the tour gives us the purchase plan of products, where the Z_{kj} be the quantity of product k purchased at market j with the purchase cost P_{kj}. The value on each edge along the tour gives the travel cost of the purchaser. Observe that, the purchaser can buy all the products with complete demand along the tour shown in Figure (3). However, this tour does not include the depot point, thus this tour is called an illegal/infeasible tour of the purchaser. One more example of a sub-tour is given in Figure (4). This tour includes the depot point. Note that, there is a possibility of purchasing a maximum of 4 units (1 unit from market 3 and 3 units from market 7) of product 1 only along this tour, and which is not meeting its actual demand of 5 units. Thus, the purchaser cannot make his complete purchase along this tour, and it is also an infeasible tour.

Table–6 contains the details of feasible patterns which are generated with PRLS, the route plan of the purchaser, and the corresponding solution. Here, $\alpha_7 = (1, 3, 7, 8, 9, 32, 35)$ be an initial feasible pattern, forms a tour $1 \to 5 \to 7 \to 2 \to 4 \to 3 \to 6 \to 1$ for the purchaser with the ordered pair of indices of letters in α_7 and provides a total cost 37 units. The total cost includes the travel cost 17 units and the purchase cost 20 units. The graphical representation of this tour and the purchase plan is shown in Figure (6). The purchaser purchase five units of product 1 each cost 1\$ at market 5, two and four units of product 2 at market 6 and market 7 each costs 2\$ and 1\$ respectively, four units of product 3 from market 7 each cost 1\$ and finally three units of product 4 at market 4 each cost 1\$.

The pattern $\alpha_6 = (1, 3, 7, 8, 9, 37)$, give a tour $1 \to 5 \to 7 \to 2 \to 4 \to 6 \to 1$ for the purchaser with an improved solution 33 units of total cost. Similarly, the pattern $\alpha_5 = (1, 3, 7, 9, 31)$, provide a better solution 31 units of total cost along the tour $1 \to 5 \to 7 \to 2 \to 4 \to 1$ of the purchaser. The search is continued further, and no improvement to the solution pattern $\alpha_6 = (1, 4, 5, 8, 18, 20)$, thus this pattern will be the optimal solution with the total cost 29 units. The total cost includes the travel cost 9 units and the purchase cost 20 units. The optimal route plan$1 \to 5 \to 4 \to 2 \to 7 \to 6 \to 1$ for the purchaser is shown in Figure (6). The purchaser

TABLE 3. Demand, availability and unit cost of products.

Product	Demand	Availability						Purchase cost per unit					
K	D_k	A_{k2}	A_{k3}	A_{k4}	A_{k5}	A_{k6}	A_{k7}	P_{k2}	P_{k3}	P_{k4}	P_{k5}	P_{k6}	P_{k7}
1	5	3	1	0	5	0	3	5	1	0	1	0	8
2	6	2	1	5	5	7	4	3	4	6	5	2	1
3	4	7	7	3	4	7	4	3	7	6	3	2	1
4	3	8	3	4	4	2	3	10	6	1	8	10	8

TABLE 4. Alphabet table.

I	C	CC	r	c	I	C	CC	r	c
1	1	1	1	5	26	4	56	7	1
2	1	2	1	6	27	5	61	1	3
3	1	3	2	4	28	5	66	1	4
4	1	4	2	7	29	5	71	3	1
5	1	5	4	2	30	5	76	3	4
6	1	6	5	1	31	5	81	4	1
7	1	7	5	7	32	5	86	4	3
8	1	8	6	1	33	6	92	4	7
9	1	9	7	2	34	6	98	7	4
10	1	10	7	5	35	7	105	3	6
11	2	12	2	3	36	7	112	6	3
12	2	14	3	2	37	8	120	4	6
13	2	16	3	7	38	8	128	6	4
14	2	18	5	6	39	9	137	2	6
15	2	20	6	5	40	9	146	3	5
16	2	22	6	7	41	9	155	5	3
17	2	24	7	3	42	9	164	6	2
18	2	26	7	6	43	∞	∞	1	1
19	3	29	4	5	44	∞	∞	2	2
20	3	32	5	4	45	∞	∞	3	3
21	4	36	1	2	46	∞	∞	4	4
22	4	40	1	7	47	∞	∞	5	5
23	4	44	2	1	48	∞	∞	6	6
24	4	48	2	5	49	∞	∞	7	7
25	4	52	5	2					

I: index, C: travel cost, CC: cumulative cost, r: row index and c: column index

purchase five units of product 1 each cost 1\$ at market 5, two and four units of product 2 at market 6 and market 7 each costs 2\$ and 1\$ respectively, four units of product 3 from market 7 each cost 1\$ and finally three units of product 4 at market 4 each cost 1\$. Although the purchase cost of both the initial feasible tour and optimal tour is same they differ in travel cost and route plan. Table –7 gives the optimal solutions and their route plans which are obtained by setting different parametric values for η and δ. It is observed that the value of η decreases the optimal solution increases where as CPU computational time is lowering.

V. COMPUTATIONAL EXPERIMENTS

This section provides the computational results of TPP with the proposed PRLS. The algorithm is coded in MATLAB programming language and tested on a PC with 1.00 GHz Intel (R) Core (TM) i5-1035G1 CPU and 4GB of RAM running the Microsoft Windows 10 operating system. Firstly, the

TABLE 5. Search table.

SN	1	2	3	4	5	6	7	V	LB	r	c	Remarks
1	1							1	22	1	5	A
2		2						2	22	1	6	R
3		3						2	22	2	4	A
4			4					3	24	2	7	R
5			5					3	24	4	2	R
6			6					3	24	5	1	R
7			7					3	24	5	7	A
8				8				4	25	6	1	A
9					9			5	26	7	2	A
10						10		6	26	7	5	R
11						11		7	27	2	3	R
12						12		7	27	3	2	R
13						13		7	27	3	7	R
14						14		7	27	5	6	R
15						15		7	27	6	5	R
16						16		7	27	6	7	R
17						17		7	27	7	3	R
18						18		7	27	7	6	R
19						19		8	28	4	5	R
20						20		8	28	5	4	R
21						21		9	29	1	2	R
22						22		9	29	1	7	R
23						23		9	29	2	1	R
24						24		9	29	2	5	R
25						25		9	29	5	2	R
26						26		9	29	7	1	R
27						27		10	30	1	3	R
28						28		10	30	1	4	R
29						29		10	30	3	1	R
30						30		10	30	3	4	R
31						31		10	30	4	1	R
32						32		10	30	4	3	A
33							33	16	36	4	7	R
34							34	16	36	7	4	R
35							**35**	**17**	**UB=37**	**3**	**6**	A
36						33		11	31	4	7	R
37						34		11	31	7	4	R
38						35		12	32	3	6	A
39							36	19	39	6	3	R
40						36		12	32	6	3	R
41						**37**		**13**	**UB=33**	**4**	**6**	**A**
--	--	--	--	--	--	--	--	--	--	--	--	--
79					26			8	33=UB			R
80				9				4	25	7	2	A
--	--	--	--	--	--	--	--	--	---	--	--	--
103					**31**			**11**	**UB=31**	**4**	**1**	**A**

TABLE 5. ***(Continued.)*** **Search table.**

--	--	--	--	--	--	--	--	--	---	--	--	--
131				21				7	31=UB			R
--	--	--	--	--	--	--	--	--	---	--	--	--
261			26					6	31=UB			R
262		4						2	22	2	7	A
263			5					3	24	4	2	A
--	--	--	--	--	--	--	--	--	---	--	--	--
266				8				4	25	6	1	A
--	--	--	--	--	--	--	--	--	---	--	--	--
277					18			6	29	7	6	A
278						19		9	29	4	5	R
279						**20**		**9**	**UB=29**	**5**	**4**	**A**
280					19			7	30>UB			R
--	--	--	--	--	--	--	--	--	---	--	--	--
331				19				7	30>UB			
--	--	--	--	--	--	--	--	--	---	--	--	--
4184			27					13	38>UB	1	3	R
4185		26						9	29=UB	7	1	R
4186	26							4	29=UB	7	1	R

TABLE 6. **Feasible and optimal solutions.**

SN	Feasible pattern	Route plan of the purchaser	Solution
1	$\alpha_7 = (1,3,7,8,9,32,35)$	$1 \to 5 \to 7 \to 2 \to 4 \to 3 \to 6 \to 1$	37(IFS)
2	$\alpha_6 = (1,3,7,8,9,37)$	$1 \to 5 \to 7 \to 2 \to 4 \to 6 \to 1$	33(IS)
3	$\alpha_5 = (1,3,7,9,31)$	$1 \to 5 \to 7 \to 2 \to 4 \to 1$	31(BS)
4	$\alpha_6 = (1,4,5,8,18,20)$	$1 \to 5 \to 4 \to 2 \to 7 \to 6 \to 1$	29(OS)
IFS: Initial feasible solution, IS: Improved solution, BS: better solution, OS: Optimal solution			

TABLE 7. **The optimal solution with respect to the varied parametric values.**

	m	n	η	δ	Optimal route	Solution (TC+PC)	Time (s)
1	7	4	7	3	$1 \to 5 \to 4 \to 2 \to 7 \to 6 \to 1$	9+20=29	0.08
2	7	4	5	2	$1 \to 5 \to 7 \to 2 \to 4 \to 1$	9+22=31	0.06
3	7	4	4	2	$1 \to 5 \to 7 \to 6 \to 1$	5+41=46	0.05
4	7	4	3	3	$1 \to 5 \to 6 \to 1$	4+49=53	0.05

performance of the PRLS tested by considering the test cases reported in Goldberg et al. [13] to compare the computational results with the available best known solutions. These test cases are classified based on the number of markets (m) ranging from 100 to 200 and the number of products (n) varying from 50 to 200 products. The test cases in Goldberg et al. [13], contain the market locations as $xy-$ coordinates and which are available in http://webpages.ull.es/users/jiriera/TPP.htm. The travel cost between a pair of market locations is considered as the Euclidean distance value and rounded to the nearest integer. For each instance, the purchase cost P_{kj} and the quantum of availability A_{kj} at market $j = 1, 2, \ldots, m$ of a product $k = 1, 2, \ldots, n$ are randomly generated in the interval (0, 1) and [1], [15], respectively. For each random number generation, a uniform distribution is used. The demand value $D_k, k = 1, 2, \ldots, n$ of a product k is obtained for different

TABLE 8. Comparative experimental results on CTPP symmetric instances with [13].

I	m	n	λ	ID	BnC	TA	$T(s)$	$\#m_1$	$PRLS$	$T(s)$	$\#m_2$	$\%gapb$	$\%gapt$
1	100	150	0.95	4	4762	4747	9	45	5235	0.12	49	9.03	9.32
2	150	50	0.9	5	5886	5879	8	63	4866	0.05	59	-20.96	-20.81
3		100	0.95	1	4456	4455	11	58	3124	0.08	30	-42.63	-42.60
4			0.99	4	2316	2314	4	25	7253	0.07	87	68.06	68.09
5		150	0.95	3	5037	5000	16	61	8969	0.13	119	43.83	44.25
6			0.99	4	2202	2201	7	26	7285	0.12	87	69.77	69.78
7		200	0.95	1	5176	5168	20	70	4384	0.16	30	-18.06	-17.88
8			0.95	3	5354	5329	16	68	9160	0.12	119	41.55	41.82
9			0.99	3	2590	2256	14	33	1891	0.16	11	-36.96	-19.30
10	200	50	0.7	4	26090	26082	127	187	14652	0.19	81	-78.06	-78.00
11			0.95	1	3963	3924	9	52	2514	0.08	18	-57.63	-56.08
12			0.95	5	3806	3792	7	51	4981	0.06	66	23.58	23.87
13		100	0.9	3	9418	9389	48	127	9045	0.06	107	-4.12	-3.80
14			0.95	2	4867	4863	22	74	4493	0.10	20	-8.32	-8.23
15			0.99	2	2790	2756	8	30	1397	0.12	12	-99.71	-97.27
16			0.99	3	2239	2224	10	28	7290	0.06	107	69.28	69.49
17			0.99	4	2564	2548	10	31	2662	0.14	29	3.68	4.28

TABLE 9. Comparison on CPU run time of PRLS verses LP reported in [14].

m	n	η	δ	Cpu_LP	Cpu_PRLS	m	n	η	δ	Cpu_LP	Cpu_PRLS
50	10	10	2	79.93	96.08	100	10	10	2	2733.03	395.11
		8	3	33.02	28.39			8	3	1179.84	105.55
		8	2	43.52	28.82			8	2	1797.24	107.62
		5	4	3.43	6.46			5	4	83.07	24.61
		5	3	5.96	6.78			5	3	152.79	28.41
	50	10	8	267.00	88.53		50	10	8	6837.62	448.711
		8	10	162.20	29.52			8	10	4595.75	166.405
		8	8	167.27	29.64			8	8	3943.25	168.89
		5	15	87.77	6.30			5	15	1394.91	35.76
		5	12	62.69	7.25			5	12	1548.44	38.11
	100	10	16	362.31	96.78		100	10	16	11883.01	606.06
		8	20	175.50	32.89			8	20	5984.45	187.44
		8	16	162.78	35.39			8	16	5479.26	195.87
		5	30	--	6.46			5	30	--	306.27
		5	24	--	7.45			5	24	---	309.45

values of $\lambda = 0.7$, 0.9, 0.95 and 0.99 using an equation (16).

$$D_k = \left\lceil \lambda \max_{j\in V} A_{kj} + (1-\lambda)\sum_{j\in V} A_{kj} \right\rceil \tag{16}$$

where λ is a control parameter, used to generate the demand value to each product k and D_k rounded to the nearest integer. If $\lambda = 0$, then $D_k = \sum_{j\in V} A_{kj}$ and interestingly in this case the tour of the purchaser must include all the markets, thus it will reduce to TSP. If $\lambda = 1$, then $D_k = A_{kj}$ and in this case, the tour of the purchaser may not include all the markets. If $\lambda < 0$, then the values stored in D_k violates equation (6) and hence the problem has no feasible solution. If $\lambda > 1$, then D_k will have negative entries which is not a realistic demand of a product. Therefore, λ can possibly assume a value in [0, 1]. Table–8 provides the comparative experimental results carried on different CTPP benchmark instances with the BnC proposed by Laporte et al. [3] and the transgenetic algorithm (TA) developed by Goldberg et al. [13].

The first four columns of Table–8 indicate the test instance, the next column BnC gives the solution with the branch and cut algorithm, the columns TA, $T(s)$, and $\#m_1$ respectively present the solution, CPU run time in seconds, and the number of markets involved in the solution tour of

TABLE 10. Experimental results on TSPLIB benchmark instances.

Instance Name	m	n	PC	TC	Int. sol.	$T(s)$	$\#m_1$	PC	TC	Opt./ best sol.	$T(s)$	$\#m_2$
SH07	7	4	20	17	37	0.08	7	20	9	29*	0.16	6
		4	35	7	42	0.07	6	35	6	41*	0.13	5
SP11	11	4	48	17	65	0.14	4	53	10	63*	0.18	3
WG22	22	10	533	426	959	1.71	11	620	114	734*	48.96	4
		10	623	406	1029	1.21	11	617	237	854*	54.04	5
DANTZING42	42	10	786	13	799	0.17	3	464	104	568	1550.6	7
		15	2454	24	2478	0.04	4	1284	252	1536	699.32	17
		20	5081	24	5105	0.04	4	2303	470	2773	549.05	30
ATT48	48	10	392	13113	13505	0.03	23	619	1742	2361*	0.14	3
		15	853	13113	13966	0.03	23	1811	1742	3553*	0.14	3
		20	947	13113	14060	0.03	23	2102	1742	3844*	0.14	3
Berlin52	52	10	1282	257	1539	0.04	4	672	474	1146	2339.2	6
		15	1293	257	1550	0.02	4	1037	393	1430	32.57	6
		20	2548	257	2805	0.04	4	1434	537	1971	151.22	9
		50	4535	257	4792	0.02	4	2690	951	3641	500.62	15
		100	10928	257	11185	0.02	4	5930	1254	7184	127.31	18
		150	14919	257	15176	0.02	4	8156	1004	9160	18.90	16
		200	20423	257	20680	0.02	4	11232	1639	12871	124.85	22
KROA100	100	50	2110	26714	28824	0.04	100	2328	3939	6267	0.06	15
		100	4485	27952	32437	0.03	100	8100	577	8677	0.10	4
		150	6313	26359	32672	0.03	100	7647	3213	10860	0.05	13
		200	8600	26942	35542	0.06	100	9763	4167	13930	0.10	15
KROA150	150	50	4713	483	5196	0.03	3	4713	483	5196	0.03	3
		100	10835	483	11318	0.03	3	10835	483	11318	0.03	3
		150	16723	483	17206	0.04	3	16723	483	17206	0.04	3
		200	9821	34169	43990	0.02	150	10529	7253	17782	0.17	29
KROA200	200	50	2687	1936	4623	0.03	23	2687	1936	4623	0.03	23
		100	5323	1573	6896	0.03	29	5323	1573	6896	0.03	29
		150	7441	3599	11040	0.03	57	7441	3599	11040	0.03	57
		200	11249	1250	12499	0.03	21	11249	1250	12499	0.03	21

PC: Purchase cost, TC: Travel cost, Int. sol.: Initial feasible solution, T(s): CPU run time in seconds, m_1: Number of markets appeared in the initial feasible tour, Opt./best sol: Optimal or best solution andm_2: The number of markets appeared in the optimal or best tour.

the purchaser with the transgenetic algorithm, the columns labeled as $PRLS$, $T(s)$ and $\#m_2$ respectively, provide the best solution, CPU run time in seconds, and the number of markets involved in the solution tour of the purchaser with the proposed modified LS. After recording the best solution with PRLS, the search was continued further up to one-hour duration. However, there is no further improvement to this best solution. Finally, the last two columns $\%gapb$ and $\%gapt$ respectively show the percent deviation between the solutions obtained by PRLS and BnC, and PRLS and TA. The percent deviation is determined using an equation (17).

$$\frac{f(PRLS) - f(BnC|TA)}{f(PRLS)} \times 100 \tag{17}$$

The observations on Table–8 are given as follows.

- A deterministic PRLS algorithm is developed to solve the proposed TPP variant optimally. The PRLS contains the simple rules of branching, bounding and termination like in branch and bound method.
- This algorithm carefully eliminates the patterns that are not feasible with the help of effective bound settings.
- The parametric values m and n are varying from 100 to 200 and 50 to 100, respectively.
- The computational results are obtained by nullifying the additional side constraint values η and δ for comparison with the results of [13].
- The PRLS obtained improved solutions to 9 test instances out of the 17 test instances considered for the experiments. This can be observed conveniently from the comparative bar plot of BnC, TA, and PRLS, shown in Figure (7).
- The number of markets involved in the best found solutions of TA and PRLS can be observed from a bar plot is shown in Figure (8).

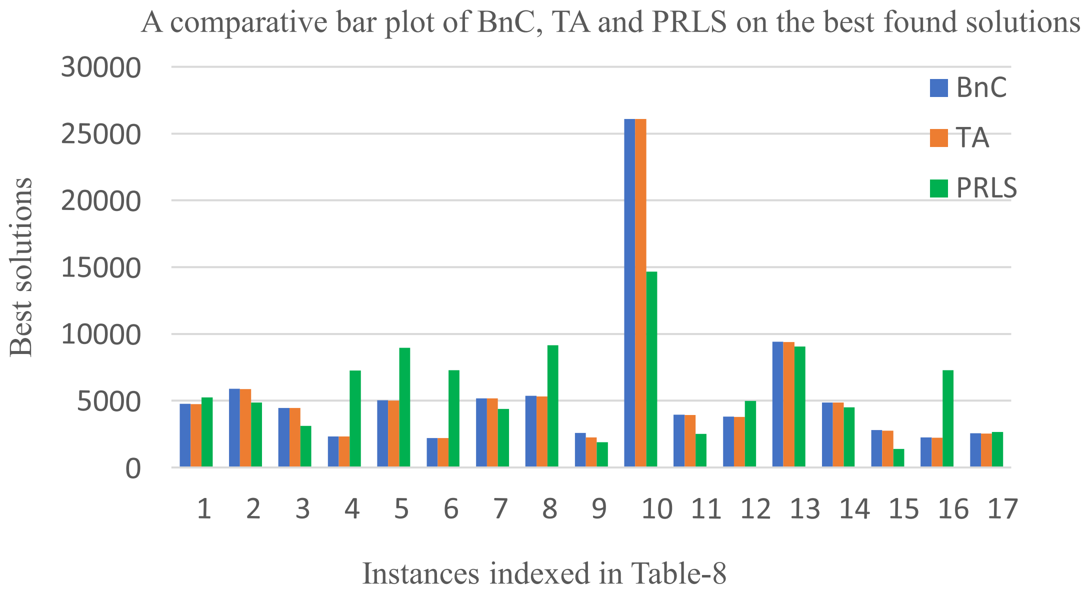

FIGURE 7. A comparative bar plot of the best found solutions.

- The improved solutions can be added to the literature for future comparisons.
- The results indicate that PRLS works efficiently to solve higher dimension instances and provide prominent solutions in reasonable computational time.
- The analysis of percent deviation shows that the proposed PRLS can obtain qualitative solutions.
- The performance of the PRLS highly depends on the different parametric values m, n, D_k, P_{kj} and A_{kj}.
- In most of the cases the improved solutions are observed on the test instances where the number of markets appear in the tour of the purchaser obtained with PRLS is lower than the number of markets appear in the tours obtained with BnC and TA.
- The average percentage gap between the solutions of BnC and PRLS is observed as -2.2158 whereas between the solutions of TA and PRLS is -0.7688. It indicates the PRLS has a good agreement with TA in providing the results.

Table–9 presents the experimental results on UTPP cases by setting different values for m, n, η and δ, recorded the PRLS average CPU run times on five independent runs of each instance and the LP relaxation results reported in [14]. On the instance size $m = 50$ both PRLS and LP relaxation have mixed results of CPU run times whereas on the instance size $m = 100$ and the PRLS CPU run times are lower than the LP relaxation. In overall CPU run times, the p-value of the t-test gives as 0.9988 under the lower tailed alternative hypothesis. This shows that the performance of PRLS is statistically superior when compared with the LP relaxation on the UTPP.

Table–10 provides the extensive computational results on some benchmark data sets. The number of markets m and the number of products n in the data sets are varying from 7 to 200 and 4 to 200 respectively. For each data set, the travel cost matrix of the purchaser is considered from TSPLIB [30] and which are available as SH07, SP11, WG22, DANTZING42, ATT48, Berlin52, KROA100, KROA150, and KROA200. The data sets SH07, SP11, and WG22 are available in TSPLIB as in matrix form and the remaining data sets are available as Euclidean coordinates. The travel cost matrix is assumed as Euclidean distance corresponds to the data sets having Euclidian coordinates. The product's availability A_{kj} and unit cost P_{kj} assumes integral values randomly in the intervals [5], [25] and [5], [30], respectively. The demand vector D_k is generated over an interval [4], [15]. For each random number generation uniform distribution is used.

For computations, a total of 30 different test data sets are used by setting different values for m and n. Performed 10 independent runs on each data set and recorded the best one. The CPU run time was recorded to find the initial solution and then the optimum or best available solution within one hour duration. When the search is completely exhausted, the latest available best solution will be the optimum solution. In some cases, the best found solution is recorded when the search was not completely exhausted. The optimum solution is obtained in eight test instances, indicating them with $*$ notation. The algorithm effectively produced the optimum solution up to the test instance size $m = 48$. It is observed that, when the instance size $m = 52$ or more, the algorithm takes more time to exhaust the search process with the implicit enumeration due to the wide range of solution space and hence recorded the best solution only. In particular, for different test cases of KROA150 and KROA200, there is no further improved solution to an initial solution within the time threshold. The time in seconds taken to produce an initial solution and the best solution of each test instance is shown

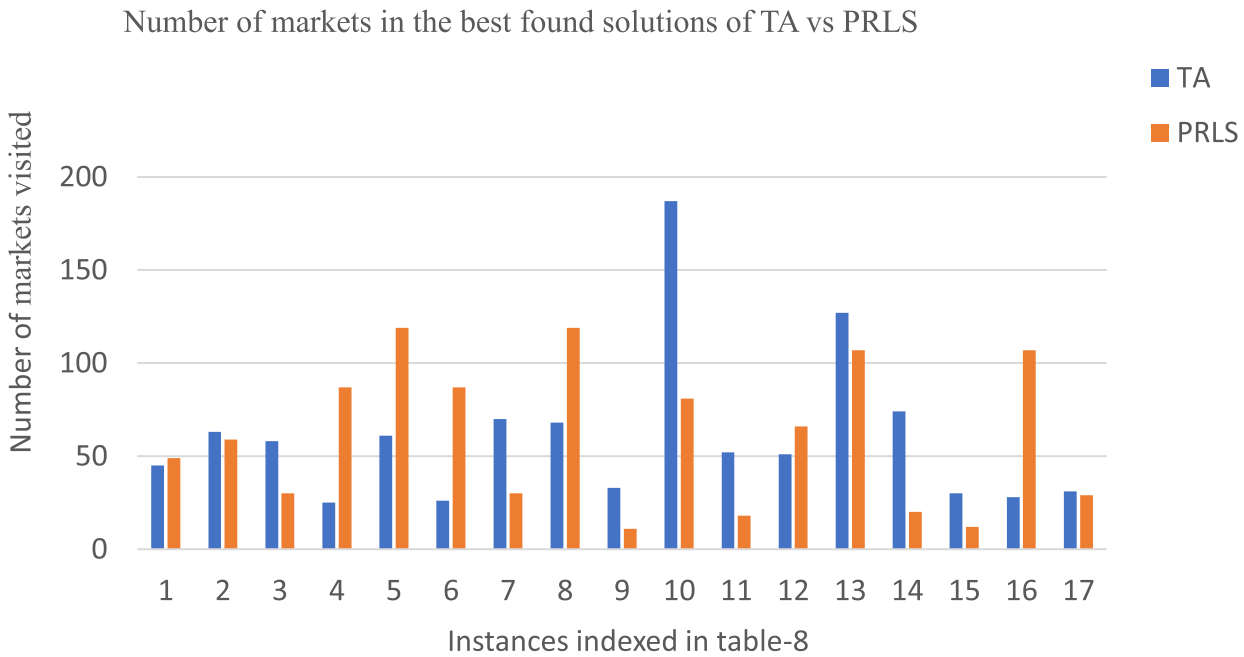

FIGURE 8. A comparative bar plot of number of markets visited.

under the column head $T(s)$. For lower dimension instances the algorithm is taking fairly less computational time than the higher dimensional instances. The number of markets present in both initial and optimal solutions are denoted by m_1 and m_2, respectively. Moreover, m_1 and m_2 may not be equal. In a few cases, the purchase cost remains the same in both initial and best found solutions but they may differ in travel cost with a different route plan. Similarly, the travel costs of the initial and best found solutions are equal yet they may differ in purchase cost with a different purchase plan. Therefore, it is important to know the purchase plan, route plan, and the key markets involved in the route plan for the purchaser for his decision making. The overall results show that the PRLS is capable in finding nearer or optimal solutions by eliminating the infeasible patterns with the help of effective branching and bounding strategies, simpler rules of feasibility and termination. However, for some of the large size data sets, the algorithm is taking more computational time to obtain the quality solutions due to the wide range of search space with more repetitive cost values appeared in the data sets.

VI. CONCLUSION

This paper presents an extended version of the variant TPP, in which the purchaser wishes to find a tour with a subset of markets that minimizes the sum of travel and purchase costs subject to satisfying a set of additional constraints. This TPP variant is formulated mathematically with integer linear programming. The TPP contains three distinct plans namely selection plan, routing plan, and purchase plan. Due to this, TPP has interesting applications in the areas of manufacturing process, scheduling of jobs to handle multipurpose multiple configuration machines, providing multiple services to different customers, routing and network designing, supply chain management, production scheduling, etc. The study contributes a deterministic PRLS to solve the TPP variant optimally and it is coded in MATLAB tool. The concepts involved in the proposed modified PRLS were explained in detail through a numerical example. The comparative experimental results carried out on different CTPP benchmark instances shows that the PRLS was found improved solutions on 9 test instances among the 17 test instances considered for which the optimum solution is not known when compared with BnC and TA. The results on UTPP show that the PRLS is statistically superior when compared with the LP relaxation.

In addition, the efficiency of the algorithm was tested on 30 more data sets, by considering the travel cost matrix from TSPLIB and generated the other parametric values involved in TPP randomly within the specified intervals. The experiments were carried out on different data sets sizes up to 200 markets with 200 products and the extensive results indicate that the algorithm effectively finds an optimal solution for lower dimension test instances, whereas it takes more time to produce the optimal solution when the size of the instance with 52 or more markets due to the wide range of the search space. The future study may include extended or modified versions of TPP with costs that are associated with the model specifications such as rebates or discounts on the purchase of products instead of existing fixed purchase costs or uncertainties on the demand, TPP with multiple depots, TPP with fast service options, TPP with multiple objectives, etc. or adding some more practical constraints into the existing model. In addition, the more sophisticated algorithms such as hybrid or evolutionary algorithms can be developed to find the prominent solutions in reasonable computational time.

ACKNOWLEDGMENT

The authors express the sincere thanks to the anonymous reviewers for their valuable suggestions and comments which help them to improve the quality of this work.

REFERENCES

[1] T. Ramesh, *Traveling Purchaser Problem*, vol. 18. New Delhi, India: Operational Research Society of India, 1981, pp. 78–91.

[2] K. N. Singh and D. L. van Oudheusden, ''A branch and bound algorithm for the traveling purchaser problem,'' *Eur. J. Oper. Res.*, vol. 97, no. 3, pp. 571–579, Mar. 1997, doi: 10.1016/s0377-2217(96)00313-x.

[3] G. Laporte, J. Riera-Ledesma, and J.-J. Salazar-González, ''A branch-and-cut algorithm for the undirected traveling purchaser problem,'' *Oper. Res.*, vol. 51, no. 6, pp. 940–951, Dec. 2003, doi: 10.1287/opre.51.6.940.24921.

[4] W. L. Pearn and R. C. Chien, ''Improved solutions for the traveling purchaser problem,'' *Comput. Oper. Res.*, vol. 25, no. 11, pp. 879–885, Nov. 1998, doi: 10.1016/s0305-0548(98)00032-x.

[5] B. Golden, L. Levy, and R. Dahl, ''Two generalizations of the traveling salesman problem,'' *Omega*, vol. 9, no. 4, pp. 439–441, Jan. 1981, doi: 10.1016/0305-0483(81)90087-6.

[6] H. L. Ong, ''Approximate algorithms for the travelling purchaser problem,'' *Operations Res. Lett.*, vol. 1, no. 5, pp. 201–205, Nov. 1982, doi: 10.1016/0167-6377(82)90041-4.

[7] F. F. Boctor, G. Laporte, and J. Renaud, ''Heuristics for the traveling purchaser problem,'' *Comput. Operations Res.*, vol. 30, no. 4, pp. 491–504, Apr. 2003, doi: 10.1016/s0305-0548(02)00020-5.

[8] A. Teeninga and A. Volgenant, ''Improved heuristics for the traveling purchaser problem,'' *Comput. Operations Res.*, vol. 31, no. 1, pp. 139–150, Jan. 2004, doi: 10.1016/s0305-0548(02)00193-4.

[9] J. Riera-Ledesma and J. J. Salazar-González, ''A heuristic approach for the travelling purchaser problem,'' *Eur. J. Oper. Res.*, vol. 162, no. 1, pp. 142–152, Apr. 2005, doi: 10.1016/j.ejor.2003.10.032.

[10] J. Riera-Ledesma and J.-J. Salazar-González, ''Solving the asymmetric traveling purchaser problem,'' *Ann. Oper. Res.*, vol. 144, no. 1, pp. 83–97, Apr. 2006, doi: 10.1007/s10479-006-0014-y.

[11] B. Bontoux and D. Feillet, ''Ant colony optimization for the traveling purchaser problem,'' *Comput. Oper. Res.*, vol. 35, no. 2, pp. 628–637, Feb. 2008, doi: 10.1016/j.cor.2006.03.023.

[12] J. Riera-Ledesma and J.-J. Salazar-González, ''Solving school bus routing using the multiple vehicle traveling purchaser problem: A branch-and-cut approach,'' *Comput. Oper. Res.*, vol. 39, no. 2, pp. 391–404, Feb. 2012, doi: 10.1016/j.cor.2011.04.015.

[13] M. C. Goldbarg, L. B. Bagi, and E. F. G. Goldbarg, ''Transgenetic algorithm for the traveling purchaser problem,'' *Eur. J. Oper. Res.*, vol. 199, no. 1, pp. 36–45, Nov. 2009, doi: 10.1016/j.ejor.2008.10.027.

[14] L. Gouveia, A. Paias, and S. Voß, ''Models for a traveling purchaser problem with additional side-constraints,'' *Comput. Oper. Res.*, vol. 38, no. 2, pp. 550–558, Feb. 2011, doi: 10.1016/j.cor.2010.07.016.

[15] R. Mansini and B. Tocchella, ''The traveling purchaser problem with budget constraint,'' *Comput. Oper. Res.*, vol. 36, no. 7, pp. 2263–2274, Jul. 2009, doi: 10.1016/j.cor.2008.09.001.

[16] E. Angelelli, R. Mansini, and M. Vindigni, ''Exploring greedy criteria for the dynamic traveling purchaser problem,'' *Central Eur. J. Oper. Res.*, vol. 17, no. 2, pp. 141–158, Jun. 2009, doi: 10.1007/s10100-008-0080-0.

[17] E. Angelelli, R. Mansini, and M. Vindigni, ''Look-ahead heuristics for the dynamic traveling purchaser problem,'' *Comput. Oper. Res.*, vol. 38, no. 12, pp. 1867–1876, Dec. 2011, doi: 10.1016/j.cor.2011.02.021.

[18] M. Batista-Galván, J. Riera-Ledesma, and J. J. Salazar-González, ''The traveling purchaser problem, with multiple stacks and deliveries: A branch-and-cut approach,'' *Comput. Oper. Res.*, vol. 40, no. 8, pp. 2103–2115, Aug. 2013, doi: 10.1016/j.cor.2013.02.007.

[19] P. Sumathi, G. Viswanatha Reddy, and P. Singamsetty, ''The travelling purchaser problem with budget constraint,'' *Int. J. Math. Trends Technol.*, vol. 66, no. 2, pp. 126–137, Feb. 2020. [Online]. Available: https://ijmttjournal.org/archive/ijmtt-v66i2p515

[20] I. Kucukoglu, ''The traveling purchaser problem with fast service option,'' *Comput. Oper. Res.*, vol. 141, May 2022, Art. no. 105700, doi: 10.1016/j.cor.2022.105700.

[21] H.-Y. Kang, A. H. I. Lee, and Y.-F. Yeh, ''An optimization approach for traveling purchaser problem with environmental impact of transportation cost,'' *Kybernetes*, vol. 50, pp. 2289–2317, Sep. 2020, doi: 10.1108/k-04-2020-0237.

[22] A. Cheaitou, S. Hamdan, R. Larbi, and I. Alsyouf, ''Sustainable traveling purchaser problem with speed optimization,'' *Int. J. Sustain. Transp.*, vol. 15, no. 8, pp. 621–640, Jun. 2021, doi: 10.1080/15568318.2020.1774825.

[23] W. N. Caballero, B. J. Lunday, and F. Meissner, ''Regulating the rebound effect in the traveling purchaser problem,'' *Eur. J. Oper. Res.*, Jun. 2022, doi: 10.1016/j.ejor.2022.06.045.

[24] Z. S. Hasanpour Jesri, K. Eshghi, M. Rafiee, and T. Van Woensel, ''The multi-depot traveling purchaser problem with shared resources,'' *Sustainability*, vol. 14, no. 16, p. 16, Jan. 2022, Art. no. 16, doi: 10.3390/su141610190.

[25] I. Kucukoglu, P. Vansteenwegen, and D. Cattrysse, ''A Tabu search algorithm for the traveling purchaser problem with transportation time limit,'' in *Computational Logistics* (Lecture Notes in Computer Science), vol. 14239, J. R. Daduna, G. Liedtke, X. Shi, and S. Voß, Eds. Cham, Switzerland: Springer Nature, 2023, pp. 248–259, doi: 10.1007/978-3-031-43612-3_15.

[26] A. Roy, S. Maity, and I. Moon, ''Multi-vehicle clustered traveling purchaser problem using a variable-length genetic algorithm,'' *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106351, doi: 10.1016/j.engappai.2023.106351.

[27] D. Manerba, R. Mansini, and J. Riera-Ledesma, ''The traveling purchaser problem and its variants,'' *Eur. J. Oper. Res.*, vol. 259, no. 1, pp. 1–18, May 2017, doi: 10.1016/j.ejor.2016.12.017.

[28] J. A. Buzacott and S. K. Dutta, ''Sequencing many jobs on a multi-purpose facility,'' *Nav. Res. Logistics Quart.*, vol. 18, no. 1, pp. 75–82, Mar. 1971, doi: 10.1002/nav.3800180106.

[29] M. S. Murthy, *A Bulk Transportation Problem*, vol. 13. Opsearch, 1976, pp. 143–155.

[30] [Online]. Available: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/

VENKATA PRATHAP DANAVULAPADU received the M.Sc. degree in mathematics from Sri Venkateswara University, Tirupati, India, in 2019. He is currently pursuing the Ph. D. degree with Vellore Institute of Technology, Vellore, Tamil Nadu, India. His research interests include operations research, combinatorial optimization, and algorithms.

PURUSOTHAM SINGAMSETTY received the Ph.D. degree in mathematics from Sri Venkateswara University, Tirupati, India, in 2011. He is currently an Associate Professor with the Department of Mathematics, Vellore Institute of Technology, Vellore. He has published more than 20 research articles in peer-reviewed journals. His research interests include operations research, combinatorial optimization, and algorithms.