

Conceptual Framework for Finding Approximations to Minimum Weight Triangulation and Traveling Salesman Problem of Planar Point Sets

Marko Dodig¹, Milton Smith²
Industrial, Systems & Manufacturing Engineering
Texas Tech University
Lubbock, Texas, USA

Abstract—We introduce a novel Conceptual Framework for finding approximations to both Minimum Weight Triangulation (MWT) and optimal Traveling Salesman Problem (TSP) of planar point sets. MWT is a classical problem of Computational Geometry with various applications, whereas TSP is perhaps the most researched problem in Combinatorial Optimization. We provide motivation for our research and introduce the fields of triangulation and polygonization of planar point sets as theoretical bases of our approach, namely, we present the Isoperimetric Inequality principle, measured via Compactness Index, as a key link between our two stated problems. Our experiments show that the proposed framework yields tight approximations for both problems.

Keywords—Computational geometry; minimum weight triangulation; combinatorial optimization; traveling salesman problem

I. INTRODUCTION

Traveling Salesman Problem (TSP), whose optimal solution is the minimum-length Hamiltonian Cycle, is the landmark problem in the field of Combinatorial Optimization. TSP is essential for the real-world applications such as vehicle routing, production planning, and design of hardware devices and computer networks. Triangulations, on the other hand, represent the most intuitive way one can partition a planar point set. They are used as valuable tools in cartography and topology of old, and mesh generations in Computer Science of new. Minimum weight triangulation (MWT) is defined as the full triangulation of the planar point set with minimal total edge length; it is also commonly referred to as the optimal triangulation. Both TSP and MWT had been proven to belong to the class of NP-hard problems. While prior research has pointed at strong links between MWT and TSP [1], we uncovered a clear knowledge gap in substantiating this relationship. This article aims to close the gap by using the tools of traditional geometry, such as the Isoperimetric Inequality principle, and proposing a conceptual framework aimed at generating close approximations to both problems.

II. BACKGROUND

A. Traveling Salesman Problem

In purely mathematical terms, TSP is the problem of finding a Hamiltonian tour of minimum weight in a complete edge-weighted graph. In our research, we consider a symmetric TSP, or STSP, in that we assume that edge-costs are symmetric, or, equivalently, that the graph is undirected. A special case of

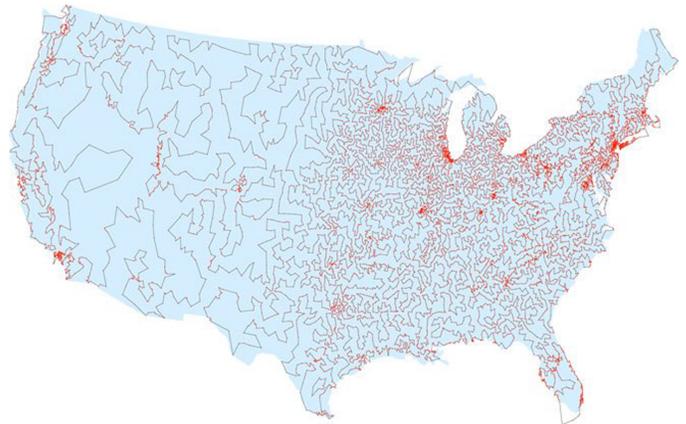


Fig. 1. TSP example of a tour through US cities [4]

the TSP is obtained when the vertices of the graph correspond to points in the Euclidean plane, and the distance between any two points is equal to the Euclidean distance between the corresponding points. The Euclidean TSP is a special case of the metric TSP, in which the costs obey the triangle inequality. Metric TSP was found to be strongly NP-hard [2]. Related to, but distinct from, the Euclidean TSP is the planar graph TSP which is the focus of our research. This is the version of the TSP in which a planar graph $G = (V, E)$ is given, with weights on the edges of E , and one seeks the minimum cost tour which uses only edges in E . Not only is this problem NP-hard, it is NP-hard even to test if a planar graph is Hamiltonian [3]. Fig. 1 illustrates a tour through all US cities with population greater than 500 as of 1998 [4].

TSP belongs to the class of NP-hard problems since no polynomial-time algorithm exists that can solve the problem optimally in polynomial time, regardless of its complexity (i.e. the number of cities in the tour). The best result to date is a solution method, introduced in 1962, that runs in time proportional to $n^2 2^n$ [5].

To quickly generate good TSP approximations, a number of heuristic algorithms has been developed over the past 50 years. Heuristic algorithms are problem-dependent techniques that use systematic procedures derived from relatively simple idea towards finding a good solution [6]. A comprehensive taxonomy of TSP heuristics is reproduced in Fig. 2.

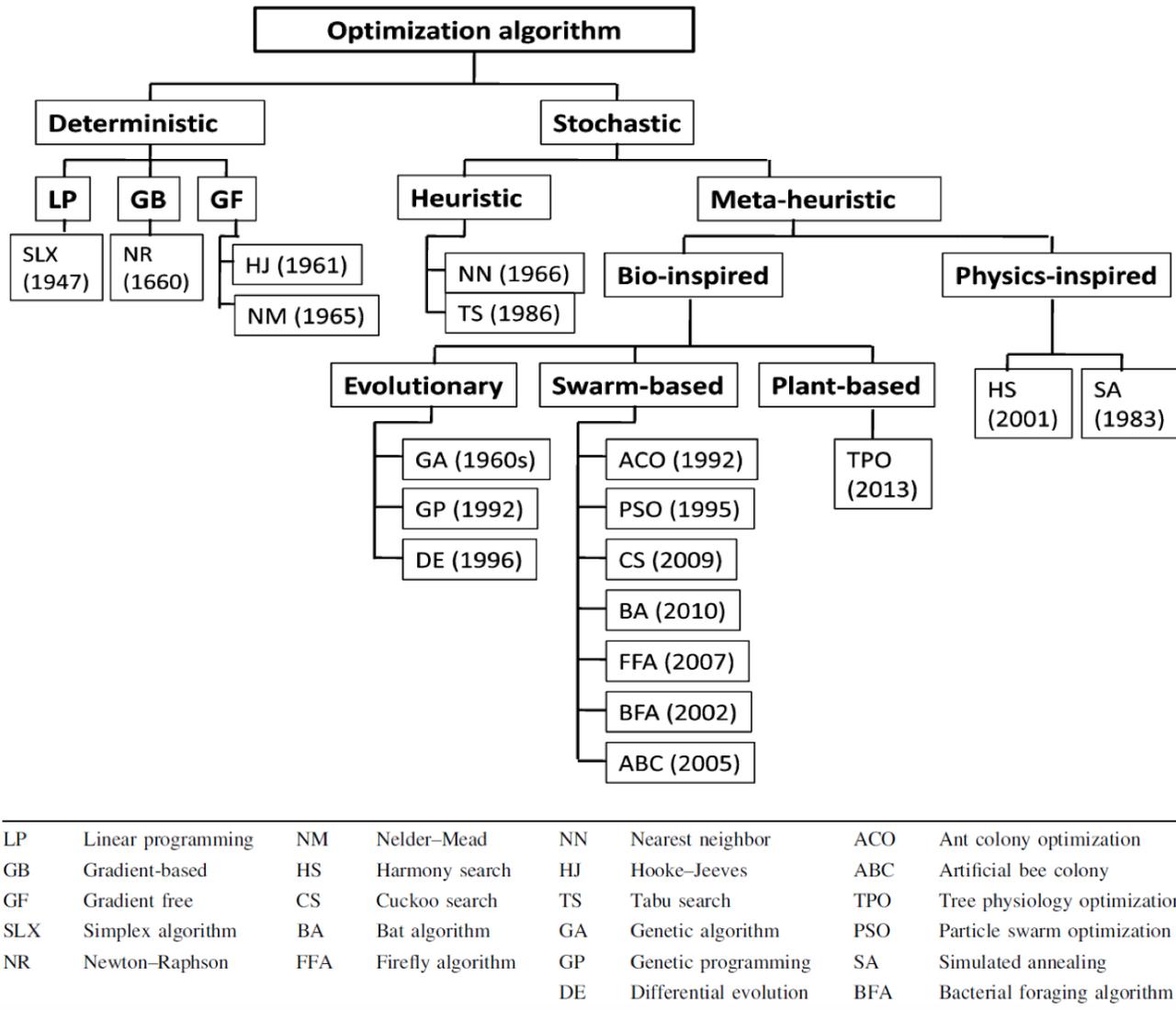


Fig. 2. Taxonomy of TSP Heuristics [6]

TSP heuristic algorithms can be divided into two distinct categories of construction and improvement. Tour construction heuristics stop when a solution is found. Improvement heuristics start with a subset of points, and then insert the rest according to some selection rules.

To summarize our problem statement, researchers have detailed out a number of innovative optimization techniques such as Simulated Annealing, Ant Colony Optimization, and Genetic Algorithms [5]; while these techniques produce good results, they do not make a dramatic shift in either the incidence of optimal solutions generated or the worst-performance guarantee. This was our main motivation to look to a geometric nature of a planar TSP.

B. Geometry and the Traveling Salesman Problem

In the early 1990s Fekete introduced TSP as one of the three optimal polygonization problems [7]. Indeed, in a planar point set S with n points, one can seek optimal

polygonizations (with all points on the perimeter representing polygon vertices) that minimize area (MINAP), maximize enclosed area (MAXAP), and minimize perimeter (TSP). In his doctoral thesis and subsequent research, Fekete had proven that both MINAP and MAXAP are also NP-hard problems and harder to solve for than TSP since edge lengths are not good representatives of the inclusion or exclusion criteria [8]. Fekete proposed a simple heuristic for minimum area polygonization, one that starts with the smallest empty 3-gon (i.e. triangle), and greedily adds to the partial polygonization candidate triangles with smallest area until full polygonization is obtained. Candidate triangles are remaining triangles that share edges with triangles already in the polygonization. Our research indicates this was the first time that triangles rather than edges were used to build a complex polygon. In his doctoral thesis a decade later, Vassilev used the area of simple triangles as a constraint, not as a quality measure, in building optimal Min-Max triangulation [9].

C. Triangulations in Computational Geometry

Triangulations represent the most intuitive way one can partition a planar point set [10]. Conceptually, triangulations have been discussed before TSP and polygonizations in general and are very valuable tools in cartography and topology of old, and mesh generations in Computer Science of new [10]. A set of triangles is called a triangulation T of the point set S if and only if: (a) every triangle of T has its vertices in S , (b) no triangle of T contains a point from S in its interior, (c) every two triangles in T have disjoint interiors, and (d) the union of all the triangles in T is exactly the convex hull of S [9]. Full triangulation completely partitions a planar point set. Triangulations of point sets in the plane have been studied for the last four decades as one of the important structures in Computational Geometry [9]. Vassilev pointed to three important attributes of a triangle that form a triangulation: (a) edge lengths, (b) angles and (c) area in his thesis, and utilized the triangle area as a constraint rather than as a general optimization criterion. This he stated was a significant motivator for his work. Our work, as it will be further revealed, is chiefly anchored on what we call the fourth triangle attribute of *Compactness Index*, which will be defined a few sections below.

The major reason to study triangulations, apart from the abundant mathematical challenges, is dictated by the practical applications [9]. Practical fields where triangulations are used include computer graphics [11], terrain approximations, multivariable analysis, numerical methods, and mesh generation [12]. Connected subgraphs of triangulations like Gabriel Graph and Relative Neighborhood Graph are used in wireless networking and ad hoc routing [13].

Triangulations are typically created to optimize some quality measure [9]. One might choose to minimize the maximum angle or maximize the minimum angle in a planar set triangulation; these are called Min-Max and Max-Min triangulations. One can also choose to minimize the total sum of edge lengths; this is called Minimum Weight Triangulation, or MWT. None of the triangulations captures imagination of researchers as much as Delaunay triangulation, an example of which is given in Fig. 3.

The origins of Delaunay dual, the Voronoi diagram, reach way back into the 17th century and writings of Descartes, who imagined the universe as a set of regions around each star and illustrated his thinking with what would be later become known as Voronoi diagrams [14]. Voronoi diagram also mimics the end stage of the cell formation, and several other key biological and chemical processes. Delaunay triangulation of a planar point set maximizes minimum triangulation angle, and contains Minimum Spanning Tree (shortest spanning tree), Nearest Neighbor Graph (graph containing edges between closest points), and Gabriel Graph (graph in which points x and y are neighbors only if there are no other points inside their diameter circle) [10].

It is therefore not surprising that researchers speculated Delaunay triangulation was also MWT, and that it also contained TSP [15]. The claim that Delaunay and MWT contained TSP was rejected by Dillencourt, who used a specific point set configuration example to disprove the claim [16]. The claim that Delaunay even approximated MWT in all point

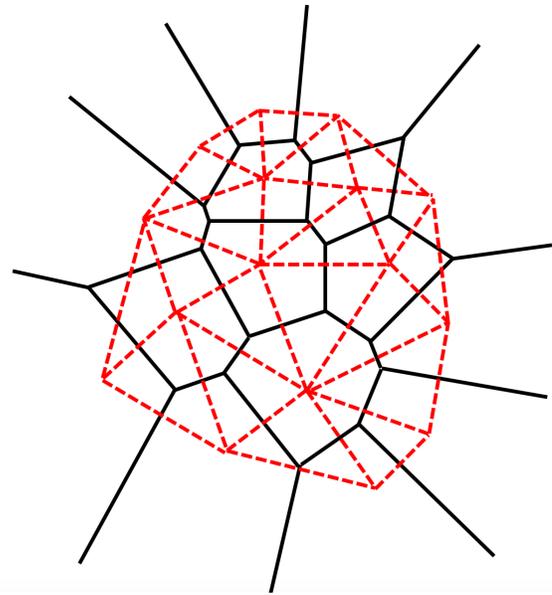


Fig. 3. Voronoi diagram (solid lines) and Delaunay triangulation (dashed lines) of a planar set of 16 points

set configurations has been rejected by Manacher and Zobrist who also used a specific point set configuration of a simple regular polygon to disprove the hypothesis [17]. However, even though Delaunay triangulation is not MWT, it does approximate it in randomized point set configurations [18]. Several researchers have since shown that using only edges from well-known triangulations and existing solution techniques like Concorde optimization engine can produce good TSP approximations, often approaching optimality. Letchford and Pearson, for instance, utilized Concorde to use edges of Delaunay triangulations in 29 TSPLIB problems to solve for TSP using Concorde [1]. They found that heuristic results were on average only 0.28% worse than optimal, while in no case being more than 3.3% worse than optimal [1].

D. Why do “Good” Triangulations Yield “Good” TSP Tours?

Our extensive review uncovered a clear knowledge gap in understanding of why TSP edges are also overwhelmingly present in MWT. To contribute to closing this gap, we researched Isoperimetric Inequality principle ($L^2 \geq 4\pi A$, where L is the perimeter and A is the area). Isoperimetric Inequality principle states that out of all geometric figures with fixed perimeter it is a circle that contains maximum area [19]. As with most of basic geometry, this special property dates to antiquity. According to the work of Kesavan, this inequality can be restated to indicate that, of all triangles with equal area, it is the equilateral triangle that has the smallest perimeter [20]. The measure of Isoperimetric Inequality can be stated as $CI = \frac{4\pi A}{L^2}$, to describe what researchers call the Compactness Index of simple geometric figures [21]. For example, any circle would have Compactness Index of 1, and all other figures have Compactness Indices of strictly less than 1, as illustrated in Fig. 4.

The remainder of this paper is organized as follows. First, we propose the conceptual framework which prioritizes empty *compact* triangles as fundamental building blocks for both

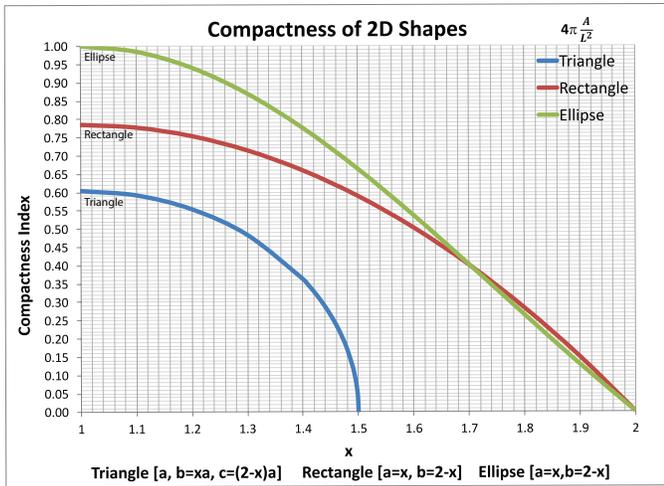


Fig. 4. Compactness Index range for 2D geometric shapes [22]

triangulations and polygonizations of choice. Second, within this proposed framework we introduce the algorithm to find approximations to MWT and TSP. Next, we evaluate the quality of the introduced algorithm experimentally. Finally, we make our conclusions and outline next steps.

III. OUR APPROACH

A. Data Hierarchy

In theoretical fields mentioned in our introduction, researchers tend to follow a traditional data hierarchy model outlined in Fig. 5a. To create TSP approximations, for instance, heuristics evaluate distances between points as key indicators of fitness for their inclusion into, or exclusion from, a solution tour. Our methodology, on the other hand, utilizes data hierarchy shown in Fig. 5b. This hierarchy organizes points into triangles, and then fits triangles into triangulations, all based on the triangle attribute of choice (i.e. Area, Perimeter, Compactness Index, Triangle Inequality). In this structure, edges are looked at only within the context of the triangles they constitute.

Similarly, polygonizations are only viewed as triangulation attributes; they are simply outer perimeters of either full or partial triangulations. This adjustment allows us to deploy system theoretical thinking in that it allows us to view polygonization

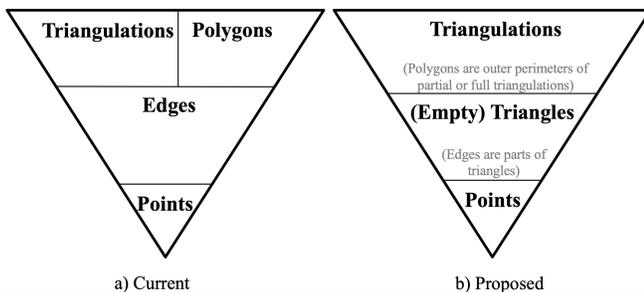


Fig. 5. a) Data hierarchy prevalent in the existing theoretical framework; b) Proposed data hierarchy guiding our research

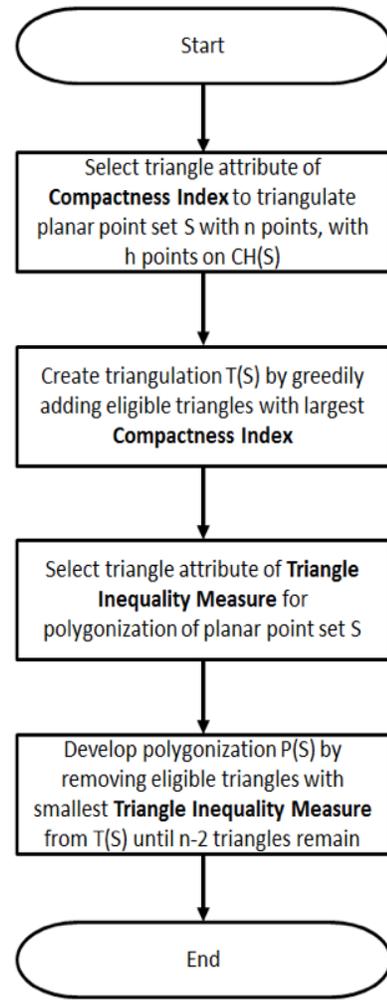


Fig. 6. Proposed Conceptual Framework

simply as a system boundary between triangles belonging to the polygonization, representing the *System*, and remaining triangles in the triangulation, representing the *Environment*. Prevalent data hierarchy presented in Fig. 5a depends on the intelligent selection of candidate edges from an exponential number of possibilities, a problem that grows more and more difficult as the number of points increases. An advantage of the proposed hierarchy lies in both the greater density of information contained in empty 3-gons and in comparatively smaller, or polynomial, number of empty triangle candidates [23].

B. Proposed Conceptual Framework

This data hierarchy ultimately allows us to propose the conceptual framework presented in Fig. 6.

Step 1 in Fig. 6 highlights the choice of *Compactness Index* as the key triangle attribute in the subsequent step of creating a full triangulation. Step 2 in Fig. 6 aims to produce a full triangulation of planar set S by greedily selecting *most compact* empty triangles. Once a full triangulation is obtained in such a way, we ensure it is also locally optimal by performing targeted edge (triangle) flips only in cases when such flips result in

shorter triangulation length. We call this triangulation *Greedy Compact Triangulation*, or GCT. We call the algorithm that creates it the GCT algorithm. We know that there are $2n - h - 2$ simple triangles in GCT of a planar set S of n points, where h represents the number of points on the $CH(S)$, or Convex Hull of S because this is a number of triangles in any full triangulation [9]. We hypothesize that because GCT favors compact empty triangles it will also minimize triangulation edge lengths; this is due to Isoperimetric Inequality principle. Step 3 in Fig. 6 highlights the choice of *Triangle Inequality Measure* as the key triangle attribute in the subsequent step of creating our TSP approximation. Step 4 in Fig. 6 aims to produce a polygon of planar set S by removing exposed triangles with minimum Triangle Inequality Measure from GCT until $n - 2$ triangles remain, since we know there are $n - 2$ simple triangles in any polygonization of planar point set of n points. The applicability of Step 6 step has been confirmed earlier [22]. In this article we focus on viability of Steps 1 and 2.

C. GCT Algorithm

GCT algorithm pseudo code is introduced in five simple steps shown below:

Input	Planar Point Set S with n points
Output	$GCT(S)$
1	Initialize array Points of size n , and load coordinates (x, y)
2	Build up <i>simpleTriangles</i> array of max $n^2 \times 6$ (3 vertices, area, perimeter, Compactness Index).
3	Sort <i>simpleTriangles</i> array (on Compactness Index; decreasing order).
4	Build <i>GCT</i> array of max $(2n - 2) \times 6$ (3 vertices, area, perimeter, Compactness Index).
5	Improve <i>GCT</i> by performing edge (triangle) flipping when a flip results in triangulation length decrease.

The time complexity of Step 1 of GCT Algorithm is $O(n)$ since there are n points in a planar point set S . Step 2 has the time complexity of at most $O(n^4)$ since there are $\binom{n}{3}$ triangles in a planar point set with n points that need to be checked against containing up to $n - 3$ remaining points. Step 3 has the time complexity of $O(3n^3 \log n)$ provided we utilize Heap Sort algorithm whose worst-case time complexity is $O(m \log m)$, where $m = O(n^3)$ and is the number of items to be sorted [24]. Step 4 has the time complexity of no more than $O(n^4)$, as we check up to $\binom{n}{3}$ candidate empty triangles in a planar point set of n points against intersecting with up to $3n - 3$ triangles already included in GCT [9]. Finally, Step 5 has the time complexity of $O(n^2)$, as this is the time complexity to transform any triangulation into another [25]. It is also easy to show that the space complexity of GCT algorithm is $O(n^3)$, since there are up to $\binom{n}{3}$ empty triangles in a planar point set of n points. In summary, GCT algorithm is polynomial in both the time and space complexity at $O(n^4)$ and $O(n^3)$, respectively.

D. Illustration

To demonstrate applicability of our approach we turn to the simple example illustrated in Fig. 7. There are $n = 52$ points in this TSPLIB planar point set called $S = berlin52$, which depicts 52 locations in the city of Berlin [26]. Triangulation lines represent edges in GCT of *berlin52*. Grey polygon with the thick red perimeter represents TSP of *berlin52*. There are exactly $h = 8$ locations on the Convex Hull of this point

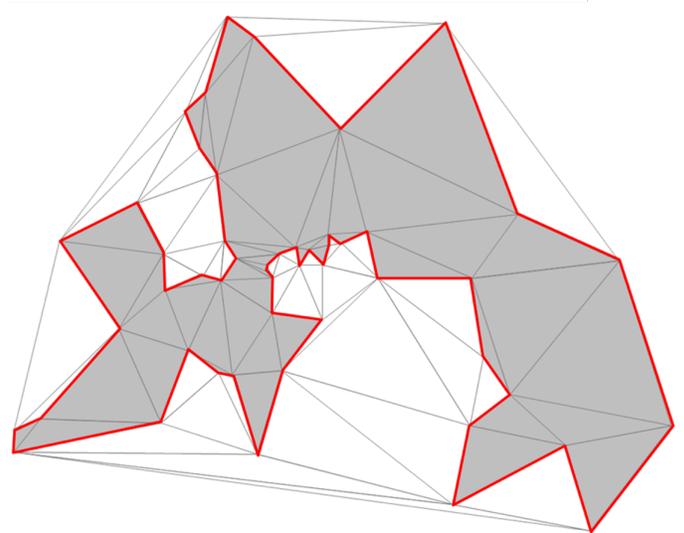


Fig. 7. TSP is fully contained within GCT for *berlin52* problem set [22]

set, and there are $n - 2 = 50$ gray triangles denoting TSP polygon. This leaves us with exactly $n - h = 44$ remaining white triangles, representing the *Environment*. One can easily see that TSP polygon is fully contained within GCT.

IV. EXPERIMENTAL METHODOLOGY

A. Hypotheses

We hypothesize that TSP polygon is fully embedded in GCT in more than 50% of the cases; this is based on our literature review [1]. In cases when full containment does not occur, we hypothesize that only a minor number of GCT triangles will be intersecting with optimal TSP, and that minimum perimeter polygon in GCT will closely approximate the optimal TSP solution, with error margins similar to those observed in research of Letchford and Pearson [1]. Finally, we speculate that best results will occur in randomized point set configurations, since prior research in restricting candidate edges to Delaunay edges worked best in these point set configurations [1].

B. Data Sets

To perform our experiments we selected 18 problem sets from TSPLIB, a well-known online problem library created to provide researchers with a broad set of test problems from various sources and properties [26]. We have chosen 11 problem sets which are given with points in general position (*att48, berlin52, ch130, eil51, eil76, eil101, gr06, gr137, rat99, rat195, rd100*). This was important as point sets in general position do not have 3 or more co-linear points. We have also chosen 7 problem sets with a significant number of co-linear points (*lin105, pr76, pr107, pr124, pr136, pr144, u159*). This was done to test performance of our framework in both point set configurations. Another reason to choose these TSPLIB problem sets was their appearance in prior research that already identified their respective MWT lengths [27].

C. Programming

To achieve our first experimental objective, we have programmed GCT Algorithm in VBA for Excel and found GCT for each of our problem sets. We have then calculated relative difference of GCT lengths to MWT lengths found in prior work of Haas [27], and identified this as GCT-to-MWT Error.

To achieve our second experimental objective, we have plotted each of the 18 resulting triangulations, together with their respective optimal TSP tours, in MATLAB. We have then visually inspected for the deviations from full TSP embeddedness in each experimental problem instance. When deviations were identified, they were removed and replaced with the most optimal edges in GCT to complete the minimum length polygons (pGCT) for each problem. Finally, we have programmed pGCT length calculations in VBA for Excel to calculate the relative difference between pGCT and optimal tour lengths, identified as pGCT-to-TSP Error.

V. EXPERIMENTAL RESULTS

Table I shows our experimental results.

On average, GCT triangulations found in our test problems are only 0.63% longer than MWT. The greatest deviation was registered with *pr107* problem set, where pGCT polygon was 4.20% longer than optimal. For 11 problem sets in general point positions the average error decreased to 0.36%. Calculated t-test statistic (one tail) at $p = 2.46\%$ shows that statistically significant difference was observed between two sets. Box plots for the entire data set, together with each type of problem sets individually, are shown in Fig. 8.

Even more impressively, pGCT polygons identified in our test problems are on average only 0.36% longer than optimal TSP solutions. The greatest deviation was registered with *pr124* problem set, where pGCT polygon was 4.78% longer than optimal. Full embeddedness was observed in 11 out of 18 cases, which represents 61.1% of the sample problems. In 5 out of 7 cases we have identified a single deviation from optimality, where the remaining 2 cases had 2 deviations from optimality. Interestingly, in one of the well-known TSP problems (*gr137*) we have identified an improvement to the stated

TABLE I. GCT AND PGCT PERFORMANCE AGAINST MWT AND TSP FOR 18 TSPLIB INSTANCES

Instance	S	n	GCT-to-MWT Error	pGCT-to-TSP Error
1	<i>att48</i>	48	0.44%	0.33%
2	<i>berlin52</i>	52	0.80%	0.00%
3	<i>ch130</i>	130	0.15%	0.05%
4	<i>eil51</i>	51	0.00%	0.00%
5	<i>eil76</i>	76	0.00%	0.00%
6	<i>eil101</i>	101	0.00%	0.00%
7	<i>gr96</i>	96	0.11%	0.13%
8	<i>gr137</i>	137	0.77%	0.08%
9	<i>lin105</i>	105	0.36%	0.00%
10	<i>pr76</i>	76	0.68%	0.26%
11	<i>pr107</i>	107	0.30%	0.00%
12	<i>pr124</i>	124	0.19%	4.78%
13	<i>pr136</i>	136	0.18%	0.00%
14	<i>pr144</i>	144	4.11%	0.02%
15	<i>rat99</i>	99	0.39%	0.00%
16	<i>rat195</i>	195	0.81%	0.00%
17	<i>rd100</i>	100	0.92%	0.00%
18	<i>u159</i>	159	1.19%	0.00%

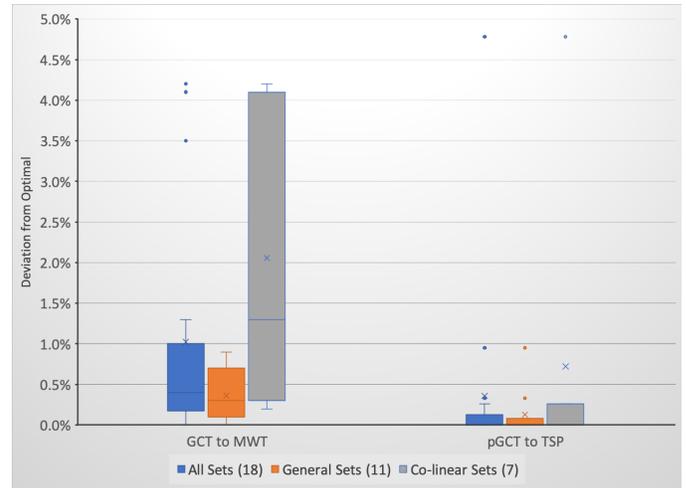


Fig. 8. Box plots for both GCT and pGCT deviations from optimal.

optimal solution. Even though the average error decreased to only 0.13%, t-test statistic (one tail) at $p = 20.86\%$ did not indicate this was a statistically significant difference between two types of problem sets. Box plots for the entire data set, together with each type of problem sets individually, are shown in Fig. 8.

VI. CONCLUSIONS

Our research proposed a novel conceptual framework, illustrated in Fig. 6, aimed at approximating both MWT and TSP. A key part of this framework is GCT algorithm we proposed to create a near-optimal TSP based on Isoperimetric Inequality principle applied to simple triangles having points from planar point sets as vertices.

We have shown that the space and time complexity of this algorithm are $O(n^3)$ and $O(n^4)$ respectively. We have also experimentally confirmed that, on average, GCT is within 0.63% of MWT in 18 TSPLIB instances. In our experiments we have also shown that GCT was at most 4.20% less-optimal than MWT. Furthermore, we have hypothesized that full TSP containment within GCT would be observed more than half of the time, and in our experimentation we have found that $pGCT = TSP$ in 61.1% of our sample problems. We have also hypothesized that the pGCT lengths would be comparable to results of Letchford and Pearson [1]. Indeed, pGCT polygons identified in our 18 TSPLIB instances were on average only 0.36% longer than optimal, with none being more than 4.78% longer than optimal.

We have also assumed that improved results will be observed in randomized TSPLIB point set configurations, which has also been confirmed in our experimentation. If we exclude 7 TSPLIB problems which have three or more co-linear points, the average observed GCT and pGCT errors were reduced to 0.36% (down from 0.63%) and 0.12% (down from 0.36%), respectively.

REFERENCES

- [1] A. Letchford and N. Pearson, "Good triangulations yield good tours," *Computers and Operations Research*, vol. 35(2), p. 638–647, 2008.

- [2] M. Garey, R. Graham, and D. Johnson, "Some np-complete geometric problems," in *STOC '76 Proceedings of the eighth annual ACM symposium on Theory of computing*. Hershey, PA: ACM, 1976, pp. 10–22.
- [3] R. Garey, D. Johnson, and R. Tarjan, "The planar hamiltonian circuit problem is np-complete," *SIAM Journal on Computing*, pp. 704–714, 1976.
- [4] E. Klarreich, "Computer scientists find new shortcuts for infamous traveling salesman problem," January 2013. [Online]. Available: <https://www.wired.com/2013/01/traveling-salesman-problem/>
- [5] W. Cook, *In Pursuit of the Traveling Salesman*. Princeton University Press, 2012.
- [6] A. Halim and I. Ismail, "Combinatorial optimization: Comparison of heuristic algorithms in travelling salesman problem," *Archives of Computational Methods in Engineering*, pp. 1–14, 2017.
- [7] S. Fekete, "Geometry and the traveling salesman problem," Ph.D. dissertation, University of Waterloo, 1992.
- [8] —, "On simple polygonalizations with optimal area," *Discrete and Computational Geometry*, vol. 23(1), pp. 73–110, 2002.
- [9] T. Vassilev, "Optimal area triangulations," Ph.D. dissertation, University of Saskatchewan, 2005.
- [10] J. De Loera, A. Rambau, and F. Santos, *Triangulation structures for algorithms and applications*. Springer-Verlag, 2010.
- [11] O. Schwarzkopf, "The extensible drawing editor ipe," in *11th Annual ACM Symposium on Computational Geometry*, 1995, pp. C10–C11.
- [12] T. Baker and P. Pebay, "A comparison of triangle quality measures," in *10th International Meshing Roundtable*, 2001, pp. 327–340.
- [13] P. Bose, P. Morin, I. Stojmenovic, and J. Urr, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7(6), pp. 609–616, 2001.
- [14] R. Descartes, *Principia Philosophiae*. Ludovicus Elzevirus, 1644.
- [15] M. Shamos and D. Hoey, "Closest-point problems," in *In 16th Annual Symposium on Foundations of Computer Science*. Long Beach, CA: IEEE Computer Society, 1975, pp. 151–162.
- [16] M. Dillencourt, "Traveling salesman cycles are not always subgraphs of delaunay triangulations or of minimum weight triangulations," *Information Processing Letters*, vol. in press, p. 339–342, 1987.
- [17] G. Manacher and A. Zobrist, "Neither the greedy nor the delaunay triangulation of a planar point set approximates the optimal triangulation," *Information Processing Letters*, vol. 9(1), pp. 31–34, 1979.
- [18] R. Chang and R. Lee, "On the average length of delaunay triangulations," *BIT Numerical Mathematics*, vol. 24, pp. 269–273, 1984.
- [19] A. Siegel, "An isoperimetric theorem in plane geometry," *Discrete & Computational Geometry*, vol. (29)2, pp. 239–255, 2003.
- [20] S. Kesavan, "From the triangle inequality to the isoperimetric inequality," *Resonance*, vol. 19(2), p. 135–148, 2014.
- [21] R. Osserman, "The isoperimetric inequality," *Bulletin of the American Mathematical Society*, vol. 8(34), p. 1182–1238, 1978.
- [22] M. Dodig and M. Smith, "Apple carving algorithm to approximate tsp from compact triangulation of planar point sets," in press.
- [23] I. Barany and P. Valtr, "Planar point sets with a small number of empty convex polygons," *Studia Scientiarum Mathematicarum Hungarica*, vol. 41(2), pp. 243–266, 2004.
- [24] J. Williams, "Algorithm 232 - heapsort," *Communications of the ACM*, vol. (7)6, pp. 347–348, 1964.
- [25] C. Lawson, "Software for c1 surface interpolation," *Mathematical Software III*, pp. 161–194, 1977.
- [26] G. Reinelt, "Tsp-lib - a traveling salesman problem library," *INFORMS Journal on Computing*, vol. 3, p. 376–384, 1991.
- [27] A. Haas, "Solving large-scale minimum-weight triangulation instances to provable optimality," in *Symposium on Computational Geometry*, 2018.