Fine-Grained Location-Free Planarization in Wireless Sensor Networks

Dezun Dong*, Yunhao Liu†§, Xiangke Liao*, and Xiang-Yang Li‡§
*National Laboratory for Paralleling and Distributed Processing,
School of Computer, National University of Defense Technology, Changsha, China;
†CSE Dept., HKUST; ‡ CS Dept., IIT, USA; §TNLIST, School of Software, Tsinghua University, China

***Abstract*—Extracting planar graph from network topologies is of great importance for efficient protocol design in wireless ad hoc and sensor networks. Previous techniques of planar topology extraction are often based on ideal assumptions, such as UDG communication model and accurate node location measurements. To make these protocols work effectively in practice, we need extract a planar topology in a location-free and distributed manner with small stretch factor. Current location-free methods cannot provide any guarantee on the stretch factor of the constructed planar topologies. In this work, we present a fine-grained and location-free network planarization method. Compared with existing location-free planarization approaches, our method can extract a high-quality planar graph, called TPS (Topological Planar Simplification), from the communication graph using local connectivity information. TPS is proved to be a planar graph and has a constant stretch factor for a large class of network instances. We evaluate our design through extensive simulations and compare with the state-of-the-art approaches. The simulation results show that our method produces planar graphs with a small constant stretch factor, often less than 1.5.**

I. Introduction

Nodes in wireless ad hoc and sensor networks are inherently placed in a geometric environment and can only communicate with nodes within a certain geometry neighborhood. The inherent geometry properties have been exploited to design a number of efficient protocols for wireless networks, such as geographic routing, topology discovery etc. Extracting a planar topology from the communication graph while preserving the intrinsic network distances is crucial for the successful execution of many protocols. As an example, in geometric routing (a.k.a geographic routing) protocols, such as GFG [1], GPSR [2], and macroscopic geographic greedy routing [3], the faces of planar communication graph are used to perform perimeter routing, which guarantees packet delivery and greatly reduces the protocol complexity. In network localization schemes, planarized network topology helps to design efficient localization algorithms [4]. In topology discovery schemes [5, 6], boundary cycles, special planar substructures of network communication graph, are extracted to locate communication holes, which contributes to the detection of faulty nodes and improves the load balancing and resilience of routing.

The most prominent approaches for addressing network planarization problem utilize the geometry locations of nodes [7–11]. In particular, the majority of those location-based algorithms [7–9] are designed under communication models of unit disk graph (UDG), and a few ones [10, 11] make an effort to construct planar graphs in quasi unit disk graph (quasi-UDG) and extended graphs. However, using the geometry locations limits the applicability of those methods because acquiring location information is often practically difficult and expensive for large-scale networks. It is usually costly to equip every node with GPS devices to get accurate location measurement. For range-based and range-free localization methods, the problem often is computationally NP-hard. Those localization algorithms usually output probabilistic results as they suffer from error accumulation, and flip ambiguity [12] etc. It is thus important to relax the assumption on location measurements to enhance the applicability of algorithms that require planar topology in resource-limited wireless networks.

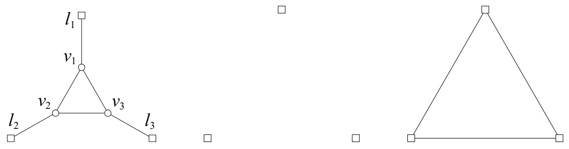
Recently, location-free planarization has received considerable attentions. Funke et al. [13] propose a distributed method to find a provable planar graph, called combinatorial Delaunay map (CDM). Zhang et al. [14] formalize network planarization as the NP-hard bipartite planarization problem, and propose a layer-by-layer planarization method. Current location-free solutions [13, 14] shed light on the challenging issue of location-free planarization. They, however, mainly focus on the objective of planarization with little consideration for the quality of constructed planar topology. Spanner property (or distortion) is widely recognized as an important metric to measure the quality of a planar subgraph [8]. A subgraph has a constant *stretch factor* (or *spanning ratio*, *dilation ratio*), if for any pair of distinct nodes, the distance in the subgraph is at most a constant times of the distance between them in the original graph. Existing location-free algorithms [13, 14] cannot provide any guarantee on the stretch factor of constructed planar topology. Zhang's method requires that network regions are of square-like shape. In an arbitrary network region with feature-rich shapes or holes, a large portion of the network cannot be planarized properly by their method and the constructed structure could have arbitrarily large stretch factor. Comparatively Funke's method does not put any conditions on the original shape of the network deployment region. Funke's method, however, outputs a well-connected CDM only when each Voronoi tile of CDM has a large diameter: their theoretical result requires the diameter be at least 290 (in hop-number metric), although their simulation results show that their method still works when the diameter of each Voronoi tile is at least 5. Under such circumstances vertices of CDM are indeed a sparse sampling of the network, which in turn results in a large stretch factor of the constructed topology. If we force vertices of CDM to be a dense sampling of the network, CDM will be disconnected into many components. In this design, we make the first attempt towards location-free planarization such that the constructed structure has a small stretch factor for most inputs. We construct a high-quality virtual planar backbone of the network, called a *topological planar simplification* (TPS), whose quality is greatly superior to CDM, as illustrated in Section V, e.g. Figure 3 (a)-(d).

The main contributions of this work are as follows. We propose a practical distributed algorithm that merely uses the connectivity information to extract a provable planar topology. We do not assume UDG model or use any location, angular, or distance information. Our method first performs a dense sampling from the original network and constructs a simplified structure that is a spanner, but maybe not planar. We then locate those edges causing non-planarity of the simplified structure, prune and modify the simplified structure to obtain a TPS while keeping the stretch factor of the modified structure as small as possible. We prove that the constructed TPS is a planar substructure for all possible inputs. We further show that, for the most instances, all the edges causing non-planarity of the simplified structure can be successfully eliminated by our method using local connectivity information, which makes our TPS have a constant stretch factor. We evaluate the performance of our method by extensive simulations. Simulations results validate that our method is effective to produce planar substructures with very small stretch factors, and robust to the network communication models. Hence, our design achieves a *fine-grained* location-free planarization. TPS has several desired features, e.g. efficient distributed construction, reflecting the topological and geometric structure of network well. As a high-quality planar graph, TPS greatly improves the performance of many applications built upon CDM graphs, e.g. recent geometric routing protocols [3, 15], can be beneficial to various applications, such as network localization [16] and segmentation [17], and topology discovery [5, 6].

The remainder of this paper is organized as follows. We discuss related work in Section II, and introduce the problem formulation in Section III. Section IV presents the algorithm of topological planar simplification. We present the evaluation in Section V, and conclude this work in Section VI.

II. Related Work

We can classify existing works on network planarization into two categories: *location-based* and *location-free*. For location-based planarization, most efforts mainly focus on finding planar structures for geometric UDGs, which is widely used in topology control for wireless ad hoc networks. Some (not completed) well known structures includes Gabriel graph (GG), relative neighborhood graph (RNG), local minimum spanning trees (LMST), restricted Delaunay graph (RDG) [7], localized Delaunay graph (LDel) [8], etc. There also exists some location-based works seeking planar graphs in quasi-UDG and extended graphs, such as CLDP [10], GridGraph [10] etc. See good surveys by Wang [18] for more location-based methods. We here pay more attention on location-free methods [13, 14] as our work is in this category.



(a) G (b) CDM of G (c) TPS of G

Fig. 1: A simple example for CDM graph

We use a simple example in Figure 1 to explain the main idea of Funke et al.'s method [13]. Given a connectivity graph in Figure 1 (a), in the first step, Funke et al.'s method selects some nodes as landmarks, denoted by squares, and other nodes are affiliated with the landmark that is closet to them in terms of in hop count. Thus, the communication graph is partitioned into disjoint subsets, called Voronoi tilings. In the example, each tile includes two nodes l_i and v_i, for i from 1 to 3. The first step itself is not new, which has been proposed and used in previous works [19, 20]. Our method also uses a similar first step. The key observation and contribution of Funke et al.'s method are the following rules to construct CDM graph: Each landmark l_i is a vertex of CDM; An edge (l_i, l_j) can be added into CDM if the following two rules are satisfied: (1) there exists a path in the graph from l_i to l_j consisting of a sequence of nodes associated with l_i followed by a sequence of nodes associated with l_j; (2) the one-hop neighbors of the path contains only nodes associated with landmark l_i or l_j. CDM is guaranteed to be planar in ρ-quasi-UDG with $\rho \geq 1/\sqrt{2}$. Funke et al.'s method, however, aims at approximating global network skeletons and considers little on the quality of CDM in terms of the important planarization metric of stretch factor. In this example, CDM only contains three isolated vertices shown in Figure 1 (b). In general CDM is often disconnected to cause large distortions when tile sizes are relatively small, as explained in Section V. Comparatively, our method targets not only planarization but also exploring effective new techniques to construct planar graph TPS in a fine-grained manner. Figure 1 (c) shows the TPS found by our method for this example.

Zhang and Jiang et al. [14] planarize a square-shaped network under realistic models with non-uniform transmission ranges. The main idea of their method is to build two specific shortest path trees and label network nodes in layers, then planarizes the network covered by the two trees in a layer-by-layer manner through formulating it as NP-hard bipartite planarization problem. Main shortages of their method are in two aspects. First, the method mainly works in regular square-shaped network, and is hard to be extended into arbitrary network regions, such as, with irregular outer boundary, complex inner holes. This is because in complex network regions the built shortest path trees would only cover a (maybe small) portion of the network, even overlap onto each other to cause faults. Consequently, the found planar graph will inevitably have a large distortion, as explained in Section V-A. Second, layer-by-layer bipartite planarization technique cannot provide any guarantee about the distortion even when the built shortest path trees can cover the whole network. Moreover, solving bipartite planarization problem essentially requires the global-scale connectivity information. They present some centralized FPT (Fixed Parameter Tractable) algorithms for the NP-Hard problem, which inevitably incurs high complexity of communication and computing. It remains unknown how to perform their algorithms in an efficient distributed manner.

III. Problem Formulation

We present network assumptions and formulate the problem of topological planar simplification. We consider a collection of nodes deployed over a plane region. The nodes are only capable of communicating with other nodes in its proximity. We assume that the coordinates of nodes are unavailable, in the sense that nodes can determine neither distance nor orientation. This makes our approach robust to situations in which geometry information is missing or only partially available. We extract planar network topology in a connectivity graph G, where vertices and edges identify the nodes and communication links, respectively.

Connectivity graph is far from a general graph in spite of missing location information, and has its inherent geometry properties. The quasi-UDG, generalized UDG model, can reflect the proximity and radio irregularity, and better captures the characteristics of wireless networks than UDG, so that it is widely used to model wireless ad hoc and sensor networks [11, 13]. A graph $H = (V, E)$ is a ρ-quasi-UDG with parameter $0 < \rho \leq 1$ if there exists an embedding $\varepsilon : V \rightarrow \mathbb{R}^2$, which maps the vertices of H into the Euclidean plane, such that for any two points u and v in H, 1) if the Euclidean distance $|\varepsilon(u)\varepsilon(v)| \leq \rho$, then (u, v) is an edge in H; 2) if $|\varepsilon(u)\varepsilon(v)| > 1$, then (u, v) is not an edge; 3) and if $\rho < |\varepsilon(u)\varepsilon(v)| \leq 1$, (u, v) can be or not an edge in H. An embedding ε is called a *realization* of H. This study uses *combinatorial* quasi-UDG, where only a collection of vertices and the neighbors of each node are known, differing with *geometric* quasi-UDG, where a realization is also given. A $1/\sqrt{2}$-quasi-UDG has an important '*link-crossing*' property. That is, given a $1/\sqrt{2}$-quasi-UDG graph H, if two edges (u, v) and (x, y) in H cross in a valid realization of H, there exist at three edges between nodes u, v, x, y in H.

Spanner property is widely recognized as an important metric to measure the quality of a planar structure [8]. Given two vertex sets X, Y of a graph H, we write $d_H(X, Y)$ as the minimum distance between any one vertex of X and any one vertex of Y on H. A general definition for graph spanner is defined as follows [21]. An (α, β)*-spanner* of H is a subgraph H' such that $d_{H'}(u, v) \leq \alpha d_H(u, v) + \beta$, for any two vertices u, v in H. If $\alpha = 1$, the spanner is called an *additive β-spanner*. If $\beta = 0$, this definition reverts to the usual definition of a *multiplicative α-spanner*, and α is the stretch factor. Unfortunately, it is theoretically infeasible to construct a planar multiplicative spanner with constant stretch factor in ρ-quasi-UDG for any $0 < \rho < 1$. (The two properties of planarity and spanner are conflict such that preserving planarity would make the subgraph have a large distortion.) To circumvent this impossibility result, we adopt the general spanner definition. Although the planar (α, β)-spanner of quasi-UDGs is still lack of enough studies, we can determinately know the existence of a planar (α, β)-spanner with bounded constant α and β for quasi-UDGs when parameter $\rho \geq 1/\sqrt{2}$. This study focuses on extracting a virtual planar backbone, topological planar simplification (TPS), from the network such that a planar (α, β)-spanner can be efficiently constructed from a TPS. We formally define topological planar simplification as follows.

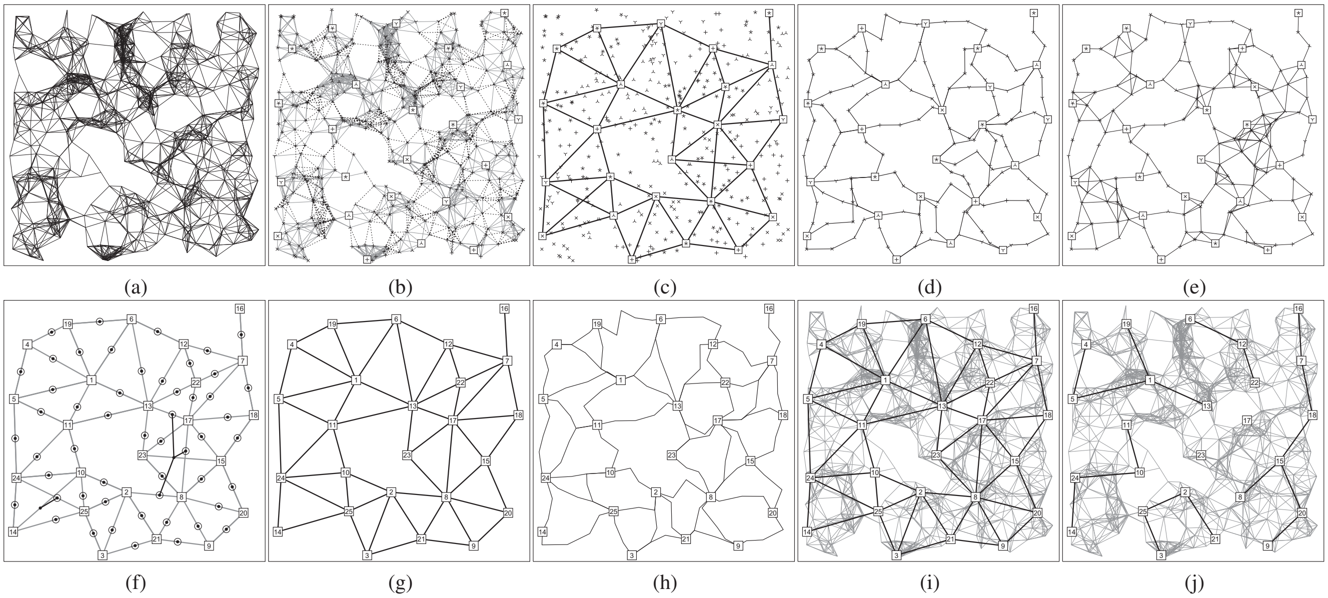
Definition 1: (*Topological Planar Simplification*) Given a connectivity graph $G = (V, E)$, a *topological planar simplification* of G is a planar graph $G' = (V', E')$, where V' is a subset of V, and each edge $(u, v) \in E'$ corresponds to a path p connecting u and v in G. G' is called as a (α, β)*-topological planar simplification* of G, if there exists two constants α and β such that $d_{G'}(u, v) < \alpha \cdot d_G(u, v)$ for $u, v \in V'$, and $d_G(v, V') \leq \beta$ for any $v \in V$.

Please note that we do not force a TPS have to be a subgraph of network connectivity graph G, which makes it more flexible and robust to construct a TPS. By default, the spanner mentioned in the rest refers to its general definition.

IV. Topological Planar Simplification Algorithm

Before describing the details of this design, we present the overview of topological planar simplification algorithm. The main idea is first to generate a constant spanner from the connectivity graph, called *restricted witness graph* (RWG). RWG is a good structure for spanner, but can be not planar. We prune and modify RWG to obtain a TPS while keeping small distortion. Our TPS protocol mainly includes three components: (1) *constructing RWG and refined underlying representation*, (2) *calculating maximal conflict-free graph*, (3) *performing conflict edge resolution*. We use an example shown in Figure 2 to explain this design. Given the connectivity graph G of a randomly deployed network, such as the one shown in Figure 2 (a), our algorithm aims at extracting a TPS graph G_{TPS} from it. The square nodes and dark-line edges in Figure 2 (i) show one found TPS by this protocol in this example.

In particular, in the first component, we construct a RWG from the original connectivity graph G, as shown Figures 2 (b) and (c). Vertices of RWG form a uniform and dense sample of the connectivity graph G. RWG is a constant spanner structure of original graph, but usually not a planar graph. We need to locate the set of edges breaching the planarity of RWG, and adjust those edges to construct a planar graph while


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 2: Topological planar simplification. (a) a random network of 0.75-quasi-UDG with 400 nodes and average degree 10.69; (b) RWG construction, the nodes in the same tile are labeled the same flag, light lines are edges among nodes in the same tile, and dot lines show edges between nodes in different tiles; (c) the RWG, whose vertices and edges are denoted by squares and lines, respectively; (d) refined underlying paths, denoted by lines; (e) the subgraph induced by all nodes in the refined underlying paths; (f) conflict-relationship graph, whose vertices and edges are dots and dark lines, respectively, the circles show a maximal independent set; (g) maximal conflict-free graph of RWG; (h) underlying pathes of maximal conflict-free graph; (i) conflict resolution for left edges and output TPS; (j) CDM graph by Funke's method.

keeping the distortion as small as possible. Without location information, it become difficult to accurately determine those edges leading to non-planarity of RWG. This design exploits the fact that the communication graph is not an arbitrary graph but having its intrinsic geometric structure derived from the underlying deployment domain. Specifically, we extract a subgraph from original graph as the refined underlying representation of RWG, and each edge of RWG corresponds to a underlying path in original graph, as illustrated in Figures 2 (d). In the second component, from the refined underlying representation of RWG, we extract a high-quality planar subgraph from RWG by combining the techniques of topological graph theory and the proximity properties of quasi-UDG. Specifically, we make a rule to determine the potential crossing underlying paths, and utilize those potential crossings of underlying paths to build a conflict-relationship graph for the RWG, shown in Figure 2 (f). Based on this conflict-relationship graph, we extract a maximal conflict-free graph from the RWG, shown in Figure 2 (g). The conflict-free subgraph of RWG is guaranteed to be a planar graph. This is because that underlying representation of conflict-free subgraph, shown in Figure 2 (h), can indeed be regarded as a planar drawing for the conflict-free subgraph through proper transformations. In the third component, for the left edges in RWG, we perform conflict resolution on them and manage to find a constant-distortion TPS graph, as shown in Figure 2 (i). Figure 2 (j) shows the CDM graph in the same network for comparison. All these procedures in TPS protocol are performed in a distributed manner using localized connectivity.

A. Constructing RWG and Refined Underlying Representation

1) Definition and Construction of RWG:

We first introduce the definition of restricted witness graph (RWG), then explain its spanner property and construction. The vertex and edge set of a graph H are referred to as $V(H)$ and $E(H)$, respectively. Let X be a vertex (or edge) set in a graph H, we use $H[X]$ to denote the vertex-induced (or edge-induced) subgraph by X. The diameter $D(H)$ of a graph G is the largest distance between any two vertices in H. RWG is inspired by the *witness complex* [19]. Witness complex becomes popular in wireless network since a special case of witness complexes, combinatorial Delaunay triangulation (CDT), is used as a tool for landmark-based routing [20]. Witness complex (or CDT), however, can generally have an arbitrary stretch factor since its tile size is not bounded. This motivates us to define restricted witness graph to achieve bounded stretch factor.

Definition 2: (*Restricted Witness Graph*) Given a graph G, and two positive constants δ and λ, a (δ, λ)-*restricted witness graph* G_R is defined on G as follows: (1) each vertex v_i of G_R corresponds to a vertex subset V_i of $V(G)$ such that $D(G[V_i]) \leq \delta$ and $\bigcup_{v_i \in V(G_R)} V_i = V(G)$; (2) an edge (v_i, v_j) of G_R exists if and only if $d_G(V_i, V_j) \leq \lambda$.

The RWG in Definition 2 is defined in a very general manner. We here present a practical construction for a specific RWG used in this work, which likes a tile-bounded CDT. We select a well-separated landmark set and partition network into proper tiles. We select a k-hops maximal independent set as landmarks set, denoted as L. Note that throughout this paper we fix k to be small constant 2 to achieve the dense network sampling. A node subset of a graph is a k-hops maximal independent set (kMIS) if it meets the following two conditions: (1) the pairwise distances of the nodes in the subset are all greater than k; (2) any extra node adding to the set will break the first condition. A kMIS can easily be computed by a greedy algorithm, which iteratively selects a node into kMIS and removes its k-hop neighborhood from the graph. A simple distributed implementation is straightforward. We then partition the graph nodes into disjoint subsets with respect to the set of landmarks selected as above. Now each node is affiliated with a landmark, which is the one in the landmark set nearest to the node and has smallest ID. Figure 2 (b) illustrates the above operations. Nodes in the same tiles are connected by lines and labeled with the same marks. Dotted lines show the edges that witness the adjacency between tiles. The edges of RWG are further obtained according to whether nodes among two tiles share a common edge, as shown in Figure 2 (c) where square landmarks are connected by bold lines to indicate the RWG. Apparently, the above construction achieves a $(2k, 1)$-RWG. Note that no ties are permitted in the initial construction of RWG, that is, two tiles not sharing nodes, which is different with CDT [19, 20]. This is because non-ties construction can eliminate some delicate edge crossing in RWG. After constructing RWG, we add all the vertices of the RWG into TPS G_{TPS} as its initial vertices. In the rest of this paper, by default we use G_R to denote the built RWG.

We analyze the characteristics of a general RWG in Definition 2. RWG is a good structure for building constant spanner, as shown in Theorem 1. Moreover, the subgraph (or tile) corresponding to each vertex of RWG has a small diameter, which makes RWG easy to build and maintain with using local connectivity. Note that RWG does not restrict that each node in a tile must be closest to the landmark in its tile as CDT, which potentially makes it more flexible to construct RWG than CDT. The built RWG, however, is often not planar in most practical instances, e.g. the one shown in Figures 2 (c). Non-planarity of RWG can be understood and explained similarly as for that of CDT, since tile-bounded CDT is a polite instance of RWG. (When the subgraph induced by each tile in a CDT has a δ-bounded diameter, the CDT is a $(\delta, 1)$-RWG.) Geometric Delaunay triangulation is a planar graph because its dual Voronoi is a plane partition and the degeneracy case of 4 points on a circle can be considered as an extremely rare or easily handled event using coordinate information. Nevertheless, the above two points are different in discrete graph settings. A partition of graph vertices does not imply a plane partition. There often exist nodes on borders of tiles that are roughly equally distant from more than three landmarks, which make critical edges between adjacent tiles appear violating the planarity of CDT.

Theorem 1: A $(2\delta+\lambda, 2\delta)$-spanner of G can be constructed from a (δ, λ)-restricted witness graph of G.

It is not difficult to show the correctness of Theorem 1. We skip the proof due to the space limitation.

2) Refined Underlying Representation:

We next define and construct a minimal connectivity subgraph to represent the RWG, called *refined underlying representation*, as follows.

Definition 3: (*Refined Underlying Representation*) A *refined underlying representation* (L, P) in G for RWG G_R is a collection of landmark nodes L and paths P. L represents the *underlying nodes* one-to-one corresponding to vertices of G_R, and P denotes *underlying paths* one-to-one corresponding to edges of G_R. Each underlying path $p_e \in P$ of $e = (l_i, l_j) \in E(G_R)$ is defined as one path in G that connects landmark l_i and l_j and consists only of nodes associated with landmark l_i and l_j.

A refined underlying representation for RWG G_R apparently exists, and can be easily found by using local connectivity. For example, a landmark node only needs to gather the connectivity within 2 hops and interacts with neighboring landmarks to construct related underlying pathes locally. Figure 2 (d) shows a refined underlying representation of RWG in Figure 2 (c). In practice, there are many candidate underlying paths for an edge of G_R. In our construction, we randomly select one shortest path to make as few as possible nodes used in the representation, so as to reduce the communication complexity. The construction of refined underlying representation for RWG is an important step to identify edge crossing in RWG and reduce the complexity of planarizing RWG. This point will be explained in the later.

B. Calculating Maximal Conflict-Free Graph

From Figure 2 (d), we can observe that some underlying paths are crossing or overlapped. In this component, with the help of underlying paths, we locate the potential crossing edges in RWG, and further extract a large planar subgraph from RWG, called *maximal conflict-free graph*, as show in Figure 2 (g). The following are important concepts defined in this component.

Definition 4: (*Contiguous*) Given two underlying paths p_1 and p_2 in G of two edges in G_R, p_1 and p_2 are *contiguous* if their distance $d_G(p_1, p_2) \leq 1$.

Definition 5: (*Edge Conflict*) Two edges e and f in G_R are of *conflict* in respect to (L, P) if their endpoints do not share one underlying node in L and their underlying paths p_e and p_f in P are contiguous, otherwise, e and f are *conflict-free* in respect to (L, P).

Definition 6: (*Maximal Conflict-Free Graph*) A *conflict-free graph* is a subgraph of RWG G_R such that any two edges in the graph are conflict-free in respect to one refined underlying representation. A conflict-free graph of G_R is *maximal* if it is not a proper subgraph of any other conflict-free graphs in G_R.

In particular, we define *contiguous* paths in Definition 4 to capture all possible crossings among underlying paths. One important observation in this work is that two contiguous underlying paths often corresponds non-crossing edges in RWG, which implies that it is over-pessimistic to simply use contiguous underlying paths to decide a pair of crossing edges in RWG. Through an in-depth analysis of underlying paths and topological properties of planar graph, we introduce the definitions about *conflict* among edges of RWG in Definition 5. We build a conflict-relationship graph for RWG, shown in Figure 2 (f), and construct a maximal conflict-free graph defined in Definition 6 form RWG, shown in Figure 2 (g). Finally we prove the planarity of a maximal conflict-free graph, whose main idea is to construct a planar drawing for the maximal conflict-free graph by proper transformation on its underlying representation, as shown in Figure 2 (h).

We now present the distributed implementation of constructing a maximal conflict-free graph G_F. Each underlying path in Figure 2 (d) can determine which underlying paths are contiguous with itself using only local connectivity. Figure 2 (e) shows the subgraph of original network graph G induced nodes in underlying paths in Figure 2 (d). The conflict relationship of two edges in G_R is determined. Hence, the conflict-relationship graph G_{cr} can be constructed from G_R, as shown in Figure 2 (f). One vertex of G_{cr} identifies an edge in G_R, and one edge of G_{cr} represents the conflict relationship of two edges in G_R, Finally, we build a maximal independent set V_{MIS} for G_{cr}, denoted by circle nodes in Figure 2 (f). We obtain an edge set E_{MIS} in G_R that corresponds to V_{MIS} in G_{cr}, and add edges E_{MIS} into G_F. As a result, G_F is a maximal conflict-free graph in G_R. Please note that G_F is maximal instead of maximum. After finding G_F, we add the edges of G_F into G_{TPS}.

We next show the planarity of a conflict-free graph in Theorem 2, whose proof are omitted here. The basic idea of the proof is to show that a planar drawing of a conflict-free graph can be constructed from its geometric realization.

Theorem 2: A conflict-free graph is planar for any quasi-UDGs with $\rho \leq 1/\sqrt{2}$.

C. Performing Conflict Edge Resolution

In this component, we dispose the left edges that cannot be selected into the maximal conflict-free graph, $E_L = E(G_R) \backslash E(G_{TPS})$. For these edges, each one conflicts with at least one edge in the maximal conflict-free graph in current underlying representation. Beside those edges really causing intersection, some conflict edges may be created due to the improper selection of underlying representation. As an example, in Figure 2 (f) the conflict between edges $(17, 8)$ and $(23, 15)$ corresponds to real edge crossing, while the fake conflict between edges $(13, 17)$ and $(23, 15)$ can be removed in other underlying representations. In this section, we present edge conflict resolution techniques for left edges due to both unfavorable underlying representations and real intersections. For those fake conflicts, we calculate new underlying paths to separate the conflicts. For real crossing conflicts, we deal with them such that stretch factor is as small as possible while the planarity is preserved.

We introduce the problem of edge conflict resolution from a simple scenario. Suppose one edge e in E_L conflicts with only one edge f in G_{TPS} regarding the current underlying representation (L, P). If we find another underlying path for e to replace its original underlying path in P, the current underlying representation is updated into (L, P'). For the new underlying representation (L, P'), if e does not conflict with f any more and still remains not to conflict with other edges, then clearly edge e can be added into G_{TPS}, and a bigger conflict-free graph is found. However, if we cannot find such good underlying path for e, the following two cases may appear: (1) a new underlying path for e causes new conflicts with other edges, or (2) the underlying path for f also needs to be modified to make e and f be conflict-free, further the modified underlying path for f can also cause new conflicts. For these cases, we wonder whether the edge e still can be added into G_{TPS} without destroying its planarity. The problem becomes more complex if edge e are conflict with multiple edges in G_{TPS}, since it becomes more difficult (or less possibility) to find an alternate path for e to eliminate all the multiple conflicts simultaneously. Ideally, it is greatly desirable if we only need to test each pair of conflict edges independently, instead of tackling all conflict edges as a whole. That is, we can add e into G_{TPS} safely if for each edge f in G_{TPS} conflicting with e, we can find a pair of underlying paths for f and e to make f and e be conflict-free. We affirmatively answer the above question by Lemma 1. If a simple condition is satisfied, called *separable-conflict testing*, we can decouple the whole conflict testing into pair testing while preserving the planarity.

We next present the definition about separable-conflict testing. We first introduce some necessary terms in graph theory and define the concept of *cycle-homotopy paths*, shown Definition 7. A *cycle* C is a subgraph of H if it is connected and each vertex in C has degree two. All the cycles of a graph form the *cycle space* of the graph if the addition of two simple cycles is defined as the symmetric difference of the two sets of edges in cycles. A *cycle basis* of a graph is a family of cycles which can span all cycles of the graph. Please refer to [22] for more conceptions on cycle basis. We can concatenate two paths with the same endpoints to obtain the *concatenation cycle*. We here only consider *simple* concatenation cycle, which is composed of the symmetric difference of edges in the two paths. The concatenation cycle of two paths sharing endpoints is either a simple cycle or a union of edge-disjoint simple cycles. For an edge e in RWG G_R, we use $U(e)$ to denote the set of all underlying paths of e in G.

Definition 7: (*Cycle-Homotopy Paths*) Given two paths p_1 and p_2 with the same endpoints in G and a positive integer ℓ_0, we say p_1 and p_2 are of ℓ_0-*cycle homotopy* in G, denoted by $p_1 \simeq p_2$, if the concatenation cycle of p_1 and p_2 admits a cycle basis in G such that each cycle in the basis has length at most ℓ_0.

Note that this work sets ℓ_0 be small constant 4, i.e. $\ell_0 = 4$.

Definition 8: (*Separable-Conflict Testing*) Given two conflict edges e_1 and e_2 in G_R, $p_1, p_1' \in U(e_1)$ and $p_2, p_2' \in U(e_2)$, let p_1 and p_2 be contiguous, and p_1' and p_2' be not contiguous. If $p_1 \simeq p_1'$ and $p_2 \simeq p_2'$, we say p_1 and p_2 are *separable paths*, and e_1 and e_2 are *separable-conflict*.

In the example in Figure 2, we perform the separable-conflict testing for the left two edges $(14, 10)$ and $(23, 15)$. The testing result shows that edge $(14, 10)$ (and resp. $(23, 15)$) is not separable-conflict with edge $(24, 25)$ (and resp. $(17, 8)$). Thus, the left edges $(14, 10)$ and $(23, 15)$ cannot be added into the current TPS G_{TPS}. It is worth noting that in this example $(23, 15)$ is separable-conflict with both $(13, 17)$ and $(2, 8)$; thus if $(23, 15)$ is selected into G_{TPS} in the previous step of maximal conflict-free graph instead of $(13, 17)$, $(2, 8)$, and $(17, 8)$; then $(13, 17)$ and $(2, 8)$ will be added into G_{TPS} in this step; this explains the effectiveness of separable-conflict testing on identifying real crossings.

After separable conflict resolution, the left edges mainly the ones causing real crossing. This means that adding each one of them into TPS G_{TPS} has the risk of destroying the planarity. This, however, is also profitable since it also means that these edges can be connected with its separable-conflict edges in the network locally. For them, we apply a simple principle of *lazy adding*. That is, our method locally tests whether these edges can be expressed (replaced with) by a path with small bounded constant μ in current TPS. If so, these edges can be deleted safely, because not adding them still makes the stretch ratio of TPS be bounded within a small constant dependent on μ. Till now, if all the left edges are disposed, our method finishes and outputs a planar graph of constant stretch factor, as described in Theorem 5.

In the example in Figure 2, the final two edges $(14, 10)$ and $(23, 15)$ can both be replaced with path of two hops in the TPS, such as paths 23-8-15 or 23-1-15 for $(23, 15)$. Thus our methods finish and output the final TPS, shown in Figure 2 (i). In this work we fix the length μ of alternate path to be a small constant 3, $\mu = 3$. This is based on the observation that in a non-separable conflict that are caused by real crossing, let $e = \{u, v\}$ be one edge in a non-separable conflict, we can mostly find an alternate path of at most three hops to connect u and v in RWG due to the link-crossing property of quasi-UDG and manner of constructing RWG. Our extensive simulations also verify that $\mu = 3$ is enough to eliminate all left edges in randomly generated networks. If unfortunately there exists an edge cannot find an alternate path bounded in a constant, more methods for non-separable conflicts can be carried out. We discuss the techniques for those special cases in Section IV-D.

We analyze the performance and prove and correctness of our algorithm. The main results are shown in Theorem 3, Theorem 4, and Theorem 5. We first present Lemma 1 to show the separable-conflict testing preserves the planarity. Following Lemma 1, we have Theorem 3. The proof of Lemma 1 is omitted here.

Lemma 1: A subgraph G_R' of G_R is planar if any two edges in G_R' are conflict-free or separable-conflict with each other.

Theorem 3: TPS is guaranteed to ba a planar graph.

We next present Theorem 4 to compare TPS with CDM analytically, whose proof is omitted here. Funke et al. [13] indicate a combinatorial Delaunay map (CDM) can faithfully reflects the topology of the network sketching, and prove a CDM has good connectivity when built from large tiles. We next show a CDM is indeed a subgraph of TPS, thus TPS has provably superior performance than CDM and inherits all good properties of CDM for large tiles.

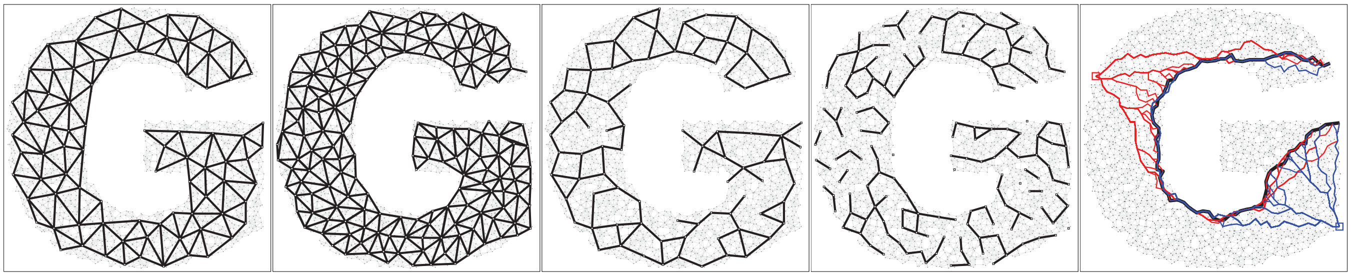
Theorem 4: TPS is a supergraph of CDM.

We then prove the distortion of the constructed graph by this design. If all the left edges are disposed through performing conflict edge resolution, our constructed planar graph is guaranteed to be $(2\mu k + 1, 2k)$-TPS, as shown in Theorem 5, whose proof directly follows Theorem 1, Theorem 2 and Theorem 3.

Theorem 5: A $(2\mu k + 1, 2k)$-TPS is obtained after successfully performing conflict edge resolution.

D. Discussion

We discuss to handle more complicated conflict resolution. We here present a simple resolution for isolated non-separable pair. The two edges that are not separable-conflict edges are called as a non-separable pair. A non-separable pair is *isolated*, if the four endpoints of non-separable pair are not the endpoints of other non-separable pair. For an isolated non-separable pair, we can locally aggregate four tiles associated with the pair into a super tile. This makes the RWG G_R be modified and isolated non-separable pair be eliminated. Clearly, in such a built super tile, a new landmark can be elected while its distance to other nodes in the super tile at most $k' = 3k + 1$ hops. Hence, after successfully eliminating isolated non-separable pair, a $(2\mu k' + 1, 2k')$-topological planar simplification is achieved. Eliminating isolated non-separable pair only use local connectivity. This mostly is effective enough since our randomly constructed RWG makes crossing edges always appear scatteredly. Our extensive simulations also verify this point. Our method can successfully find TPS without failure cases when building RWG by randomly selected landmarks, thus are practically effective. Conceptually, it is possible to artificially construct a RWG instance to make the non-separable crossing hinged into a large components of network scale by elaborate deployment and selection of nodes. For such ill cases, isolated pair conflict resolution is not available to guarantee the planarity. Indeed, it becomes impossible to solve such ill cases using only local connectivity information. If forcing to find planar spanner for such an ill RWG graph, we have to collect connectivity information of network scale. It is too cost for large networks. It is more feasible and efficient to perform our method on such an artificially ill RWG as follows. We can locate the part of ill network regions where large hinged crossing components happens. We repartition this ill part of networks



(a) TPS, $k = 3$ (b) TPS, $k = 2$ (c) CDM, $k = 3$ (d) CDM, $k = 2$ (e) ZJC

Fig. 3: (a)-(d) compare TPS and CDM with varying tile size k. Average node degree of the network is 8.51. (e) shows the results of ZJC including two trees rooted at two square nodes and *base path* [14] denoted by the bold line.

by randomly selected landmarks, and build a new substituted RWG. The new RWG can make ill scenarios disappear with high probability. As a result, we can successfully planarize the ill network regions in a divide-and-conquer and convergent manner.

V. Evaluation

We conduct extensive simulations to evaluate the effectiveness of this approach. By varying the tile sizes, node density, communication models and deployment region, we evaluate TPS with regard to stretch factor and robustness of planarity. We compare this design with two state-of-the-art approaches: CDM graph proposed by Funke and Milosavljevic (denoted as CDM) [13], and robust planarization proposed by Zhang, Jiang and Chen (denoted as ZJC) [14]. They are currently two best distributed methods using solely node connectivity to achieve network planarization.

A. Qualitative Evaluation

We first present qualitative simulations to visually demonstrate the quality of our approach, and explain the large distortions of CDM and ZJC. In this set of simulations, nodes are randomly deployed in regions of different shapes and with varying density. By default the networks are generated under 0.8-quasi-UDG model.

We compare our TPS with the CDM with changing the parameter k for kMIS landmarks to adjust tile sizes, $k = 2, 3$. Figure 3 shows a set of results in a 'G'-shape network. Compared with TPS, CDM is rather sparse and has a large distortion, and is disconnected into many connected branches especially when k becomes small. We next examine the results of ZJC in the same network, shown in Figure 3 (e). We see that two shortest trees built by ZJC only cover a small portion of the network, which causes large distortions among nodes that are not covered by the trees. (Please note that two trees locate at the same side of base path, which is also unexpected by ZJC). By varying different shapes, we find ZJC cannot work well in the following two scenarios: (1) the networks with only one dominating long path, which makes the networks have not two orthogonal long paths. Figure 3 belongs to such case. (2) the networks have narrow bottlenecks in the center region, which often makes the built shortest trees only cover a degenerate line-shape region. Due to the page limit, we skip more results on CDM and ZJC in network fields of various different shapes. Generally, we have the following observations: TPS and CDM are independent with the network shape since they only using local connectivity information, while ZJC needs global connectivity and probes to network shapes. Hence, we will mainly compare TPS with CDM in the next quantitative results.

B. Quantitative Results

We then quantitatively examine the distortion and robustness of TPS in random networks. We deploy 2500 nodes in a square area by uniformly random distribution. Under each configuration, our simulation takes 100 runs with random network generation, and we report the average.

We have shown that the successful implementation of our method will guarantee the stretch factor bounded in a small constant. We here examine the practical stretch in random networks, including the worst stretch and the average. The worst stretch is the maximum stretch between any two nodes in a network. In this set of simulations, we test stretch factor by changing the densities of k-MIS landmarks, from $k = 2$ to 6, and fixing 0.8-quasi-UDG graph model, where the average node degree 11.

We first measure the multiplicative stretch for any two landmarks in a TPS graph. TPS is regarded as an edge-weighted graph. For each edge (l_i, l_j) in the TPS, its weight is set to be the hop number of the shortest underlying path for the edge. The stretch for any two landmarks l_i and l_j in a TPS graph is the ratio of their distance $d_T(l_i, l_j)$ in TPS to distance $d_G(l_i, l_j)$ in original network. Table I shows the results, where t_{max} and t_{avg} denote the worst and average stretch factor, respectively, and σ is the standard deviation. The columns of α and α, β show the theoretical bound for the stretch, presented in Theorem 5. In all these testings, our method achieves a very small stretch factor. Specifically, the worst and average stretch factors are bounded in 2 and 1.2, respectively, which are much better than the theoretical bound given by Theorem 5.

We next check the multiplicative stretch for any two nodes in the whole network. To calculate the multiplicative stretch, we construct a spanning graph G' of G from a TPS. In

TABLE I: Planarization Results of TPS

	Stretch for landmark nodes					Stretch for nodes in the whole network					Comparison with CDM					
k	α	t_{max}	σ	t_{avg}	σ	α, β	t_{max}	σ	t_{avg}	σ	$branch$	σ	$degree$	σ	$ratio$	σ
2	13	1.931	0.035	1.194	0.033	13,4	7.621	0.231	1.406	0.104	36.533	2.271	0.783	0.018	4.413	0.102
3	19	1.830	0.046	1.171	0.033	19,6	8.115	0.511	1.468	0.137	2.533	0.427	1.264	0.027	2.495	0.055
4	25	1.693	0.034	1.154	0.034	25,8	8.603	1.104	1.527	0.168	1.167	0.126	1.536	0.025	1.848	0.032
5	31	1.608	0.029	1.139	0.035	31,10	9.278	1.114	1.601	0.203	1.033	0.061	1.683	0.033	1.495	0.032
6	37	1.518	0.050	1.120	0.034	37,12	9.987	1.139	1.659	0.225	1.033	0.061	1.739	0.041	1.277	0.026

particular, for any two nodes u and v in G, their distance $d_{G'}(u,v)$ in G' is composed of three parts: $d_{G'}(u,v) = d_G(u,l_u) + d_T(l_u,l_v) + d_G(l_v,v)$. l_u (or l_v) is the landmark of the tile that u (or v) belongs to.

From Table I, we can see average and worst stretches are also bounded by small values that are greatly better than our theoretical bounds. Further, we can find that a TPS of smaller tile size k provides a better multiplicative stretch for the whole nodes. This is consistent with our intuition that a TPS with small tiles achieve a dense sampling for the network and produces a spanner with a small addictive stretch, thus can better reflect real network distances.

We also compare TPS with CDM in the same networks. We find that in each network configuration there always exist some randomly generated instances where CDM is disconnected, which makes use be not able to calculate the average stretch factor for CDM. Hence, for CDM we check its number of connected branches and node degrees. In Table I, $branch$, $degree$ and $ratio$ denote the number of connected branches, vertex degree of CDM, and the ratio of edge number of TPS to that of CDM, respectively. From Table I, we can see that TPS contains much more edges than CDM, and has much better connectivity than CDM, especially for small tile size k.

VI. Conclusions

As a crucial issue in wireless ad hoc and sensor networks, network planarization is previously addressed either under ideal assumptions, or in relaxed models while not providing any guarantee on the quality in terms of connectivity and distortion. We present a practical method to perform topological planar simplification on networks, and take the first attempt towards extracting a provably planar topology from the network in a fine-grained location-free manner. Our constructed graph is proved to be a planar graph and has a constant stretch factor for a large class of network instances. The simulation results show that our method will produce planar graphs with a small stretch factor, which significantly outperforms the state-of-the-art approaches.

VII. Acknowledgments

The authors are grateful for a variety of valuable comments from the anonymous reviewers. This work is supported in part by National Basic Research Program of China (973 Program) under Grants 2010CB328000 and No. 2011CB302705, NSFC under grants No. 60621003, No. 60903223, No. 60903224, and No. 60828003, NSF CNS-0832120, NSF CNS-1035894, program for Zhejiang Provincial Key Innovative Research Team, program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program).

References

[1] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. of ACM DIALM*, 1999.

[2] B. Karp and H. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. of ACM MobiCom*, 2000.

[3] S. Funke and N. Milosavljevic, "Guaranteed-delivery geographic routing under uncertain node locations," in *Proc. of IEEE INFOCOM*, 2007.

[4] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. of ACM SenSys*, 2004.

[5] Y. Wang, J. Gao, and J. S. Mitchell, "Boundary recognition in sensor networks by topological methods," in *Proc. of ACM MobiCom*, 2006.

[6] D. Dong, Y. Liu, and X. Liao, "Fine-grained boundary recognition in wireless ad hoc and sensor networks by topological methods," in *Proc. of ACM MobiHoc*, 2009.

[7] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu, "Geometric spanner for routing in mobile networks," in *Proc. of ACM MobiHoc*, 2001.

[8] X. Li, G. Calinescu, and P. Wan, "Distributed construction of a planar spanner and routing for ad hoc wireless networks," in *Proc. of IEEE INFOCOM*, 2002.

[9] X. Li, W. Song, and W. Wang, "A unified energy efficient topology for unicast and broadcast," in *Proc. of ACM MOBICOM*, 2005.

[10] Y. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic routing made practical," in *Proc. of USENIX NSDI*, 2005.

[11] J. Chen, A. Jiang, I. Kanj, G. Xia, and F. Zhang, "Separability and topology control of quasi unit disk graphs," in *Proc. of IEEE INFOCOM*, 2007.

[12] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[13] S. Funke and N. Milosavljevic, "Network sketching or: How Much Geometry Hides in Connectivity?–Part II," in *Proc. of ACM-SIAM SODA*, 2007.

[14] F. Zhang, A. Jiang, and J. Chen, "Robust Planarization of Unlocalized Wireless Sensor Networks," in *Proc. of IEEE INFOCOM*, 2008.

[15] R. Sarkar, X. Yin, J. Gao, F. Luo, and X. Gu, "Greedy routing with guaranteed delivery using ricci flows," in *Proc. of ACM/IEEE IPSN*, 2009.

[16] S. Lederer, Y. Wang, and J. Gao, "Connectivity-based Localization of Large Scale Sensor Networks with Complex Shape," in *Proc. of IEEE INFOCOM*, 2008.

[17] X. Zhu, R. Sarkar, and J. Gao, "Shape segmentation and applications in sensor networks," in *Proc. of IEEE INFOCOM*, 2007.

[18] Y. Wang, *Wireless Sensor Networks and Applications*. Springer, 2008, ch. 5,Topology Control for Wireless Sensor Networks, pp. 113–147.

[19] G. Carlsson and V. de Silva, "Topological approximation by small simplicial complexes," in *Preprint*, 2003.

[20] Q. Fang, J. Gao, L. Guibas, V. de Silva, and L. Zhang, "GLIDER: Gradient landmark-based distributed routing for sensor networks," in *Proc. of IEEE INFOCOM*, 2005.

[21] S. Baswana, T. Kavitha, K. Mehlhorn, and S. Pettie, "New constructions of (α, β)-spanners and purely additive spanners," in *Proc. of ACM-SIAM SODA*, 2005.

[22] C. Liebchen and R. Rizzi, "Classes of cycle bases," *Discrete Applied Mathematics*, vol. 155, pp. 337–355, 2007.