Expert Systems With Applications 80 (2017) 311–322

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Improved search methods for assessing Delay-Tolerant Networks vulnerability to colluding strong heterogeneous attacks

Doina Bucur[a,*], Giovanni Iacca[b]

[a] University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands
[b] INCAS[3], Dr. Nassaulaan 9, 9401 HJ Assen, The Netherlands

ARTICLE INFO

Article history:
Received 21 December 2016
Revised 14 March 2017
Accepted 15 March 2017
Available online 21 March 2017

Keywords:
Security
Mobile
Evolutionary algorithm

ABSTRACT

Increasingly more digital communication is routed among wireless, mobile computers over ad-hoc, unsecured communication channels. In this paper, we design two stochastic search algorithms (a greedy heuristic, and an evolutionary algorithm) which automatically search for strong insider attack methods against a given ad-hoc, delay-tolerant communication protocol, and thus expose its weaknesses. To assess their performance, we apply the two algorithms to two simulated, large-scale mobile scenarios (of different route morphology) with 200 nodes having free range of movement. We investigate a choice of two standard attack strategies (dropping messages and flooding the network), and four delay-tolerant routing protocols: First Contact, Epidemic, Spray and Wait, and MaxProp. We find dramatic drops in performance: replicative protocols (Epidemic, Spray and Wait, MaxProp), formerly deemed resilient, are compromised to different degrees (delivery rates between 24% and 87%), while a forwarding protocol (First Contact) is shown to drop delivery rates to under 5% — in all cases by well-crafted attack strategies and with an attacker group of size less than 10% the total network size. Overall, we show that the two proposed methods combined constitute an effective means to discover (at design-time) and raise awareness about the weaknesses and strengths of existing ad-hoc, delay-tolerant communication protocols against potential malicious cyber-attacks.

1. Introduction

Delay-Tolerant Networking (DTN) is a family of multi-hop communication protocols suitable for applications which have discontinuous connectivity (Fall & Farrell, 2008), such as terrestrial-and-space networks where some of the nodes are Low-Earth Orbiting Satellites (Jain, Fall, & Patra, 2004), or the more recent opportunistic urban networks (Burgess, Gallagher, Jensen, & Levine, 2006). Since the connectivity graph of such a network is dynamic and often disconnected, for a message to be communicated along on an end-to-end path between any two nodes *A* and *Z*, the message may be stored in the buffers of any number of intermediate nodes *B, C*, etc., until an opportunity for further delivery towards *Z* arises at *B* and *C*. DTNs are an option for the future of the Internet in communication-challenged communities, where they add a degree of connectivity, and for local, city-scale networking which does not rely on a preexisting communication infrastructure.

By definition, urban DTN applications consist of volunteer contributors, and must remain *open* to all willing participants, rather than require authentication. Thus, all nodes in an urban DTN are effectively insiders. One cyber-security question therefore arises: will a DTN communication protocol be resilient to insider attacks, in various practical, mobile deployment scenarios? It is easy to design attacks where a malicious insider *B* either drops the messages it should route, or injects an unusually large number of messages into the network. In the worst case, a group of *strong colluding* attackers, i.e. collaborative malicious nodes with full knowledge of the network, may disrupt the network. If the urban DTN normally propagates news about crime and accidents in the city, the actions of malicious attacker could raise the number of victims. For an analytical answer to this question, one should calculate the *optimal* set of nodes to be attacked; however, this computation is hard: even given a known node connectivity pattern, this vertex vulnerability problem was proven NP-hard (Burgess et al., 2006). In related work, the calculation was done using simple *random sampling*, and also *greedy heuristics* with relatively few variables to optimize (e.g., the attackers did not have freedom of movement) (Burgess, Bissias, Corner, & Levine, 2007; Kempe, Kleinberg, & Éva Tardos, 2015). These studies showed that small DTNs were remarkably resilient

* Corresponding author.
E-mail addresses: d.bucur@utwente.nl (D. Bucur), giovanniiacca@incas3.eu (G. Iacca).

http://dx.doi.org/10.1016/j.eswa.2017.03.035
0957-4174/

against a large number of attackers: e.g., the data delivery rate in a certain DTN scenario dropped to 50% only when 50% of the nodes in the network executed a single attack logic. In our own previous work (Bucur, Iacca, Gaudesi, Squillero, & Tonda, 2016; Bucur, Iacca, Squillero, & Tonda, 2015) we empirically found these results by using evolutionary algorithms (EAs), yet also with a low degree of generality (in particular, our previous experiments were limited to fixed-size groups of *homogeneous* attackers, i.e. malicious nodes constrained to a single attack logic). In this paper we further develop this research line by presenting the following contributions:

- We design two *improved* search methods to find the most effective attack strategies upon a given DTN communication protocol:
  1. A **greedy heuristic** inspired by the existing literature (Kempe et al., 2015), but improved here —in an original way— to allow the attackers full freedom of movement.
  2. An **evolutionary algorithm** that optimizes a large set of variables concerning the attack strategy, namely:
     - the number of attackers in the group;
     - the attack logic, independently for each attacker in the group;
     - the speed and route of each attacker in the group.

     By allowing the EA to choose the number of attackers and the attack logic for each attacker, we largely improve upon our previous works (Bucur et al., 2016; Bucur et al., 2015) as we can now find optimally-sized groups of *heterogeneous* attackers (as opposed to fixed-size homogeneous groups).
- We experiment via simulation on two realistic, large-scale urban DTN scenarios with completely different map morphology: a regular, grid-like 5 km^2 area of downtown San Francisco, US, and an irregular (from a morphological point of view) 5 km^2 area of Venice, Italy. For each scenario we consider four DTN routing protocols, i.e. First Contact, Epidemic, Spray and Wait, and MaxProp. This set of protocols was chosen because it covers the range of features specific to DTNs, in particular the aspect of message replication (discussed later in Section 2.1). Also, we study the presence of a variable-sized group of attackers (from 1 to 20), out of a total network size of 200 mobile nodes (while related work focuses on much smaller networks). In all the test cases, we compare the greedy heuristic and the EA against random sampling.
- We find that both our improved search methods shed new light upon the weaknesses of these protocols now known from existing literature: replicative protocols (Epidemic, Spray and Wait, MaxProp), deemed resilient in the related work, are found compromisable to very different degrees (down to delivery rates of 24%, 68%, and 87%, respectively), while a forwarding protocol (First Contact) is shown to drop its delivery rate under 5% — in all cases by well-crafted attack variables, but with an attacker group of size less than 10% the total network size.

The rest of the paper is organized as follows. Section 2 details the background concepts. Section 3 introduces the related work on search algorithms applied to DTN security. The proposed methods are described in Section 4, and the numerical experiments are presented and analysed in Sections 5 and 6, respectively. Finally, Section 7 concludes this work.

2. Background

In this section we first summarize the main elements of DTN routing, and the existing protocols. We then focus on the computational models of agent movement in DTNs. Finally, we detail the most common kinds of attack in these networks.

2.1. DTN routing: routing and performance objectives

The main functional performance factor for a Delay-Tolerant Network is the *data delivery ratio* (DDR): the percentage of those honest messages injected in the network which were successfully delivered to their destinations. We focus our study on optimizing the effect of attackers upon data delivery, i.e., on minimizing the DDR. Other relevant performance factors for DTNs include *message latency*, i.e., the average time interval between the injection of a message in the network (by an honest node) until the delivery of the message.

In the design of a DTN routing protocol, the computation of routing paths at a node may be aided if the node is able to predict future networks factors such as the pattern of contact with other nodes, the set of nodes with congested buffers, and the pattern of traffic demands. While such network knowledge may be acquired in practice by an attacker via monitoring the network, many practical protocols do not assume any, and are thus *zero-knowledge*.

DTN protocols differ in their choices for *message replication*: replicative DTN routing protocols are designed such that nodes will inject into the network copies of each message they need to forward, and each of these copies is forwarded independently. This is a best-effort scheme to raise the likelihood that a copy of the message reaches its destination, given unpredictable mobility in the network. On the other hand, in single-copy forwarding protocols only one copy of a message exists at each time in the network. A survey of existing protocols is given in Wei, Liang, and Xu (2014).

We describe classic DTN routing protocols in these two categories, below.

First Contact (Jain et al., 2004) is a zero-knowledge protocol, purely forwarding rather than replicative, and thus economical; it routes messages opportunistically using any available contacts with other nodes. A single copy of each message in the network exists at a time, is stored in a node's finite buffer, and is forwarded to the first available contact (if more contacts are available, one is chosen randomly among all the current contacts). If a node's buffer is already full with other messages, any new messages are dropped. Due to the lack of route planning, no guarantee can be given about the likelihood of a message reaching its intended destination. On simple network topologies, First Contact was shown (Jain et al., 2004) to have performance comparable to partial-knowledge protocols; the performance will degrade in complex topologies to varying degrees, depending on the network load.

Epidemic routing (Vahdat, Becker et al., 2000) is zero-knowledge and essentially adapts the basic concept of network flooding to DTNs. Every node carrying a message will replicate this message to a number of carrier nodes. The amount of replication is limited by two fixed upper bounds: on (a) the message *hop count* and (b) the *buffer space* that any node dedicates to storing the messages of any other nodes. When two nodes come into contact, one requests copies of those messages from the other's buffer which it has not seen yet, subject to the hop-count limit. When the per-node buffer becomes full, the oldest messages are dropped. In a scenario with 50 nodes visiting random points in a 1500 m $\times$ 300 m area, it was shown that (1) lowering the message hop count can preserve the delivery ratio, but will raise the average message delay, and (2) the delivery rate reaches 100% when the buffers can store more than 10% of all the messages injected in the network (Vahdat et al., 2000).

Spray and Wait (Spyropoulos, Psounis, & Raghavendra, 2005) combines replicative and single-copy features. At every injection of a new message into the network (the "spray" phase), a number n of messages copies are spread to different carrier nodes. If the destination node of the message is not found among these carriers, then the n carriers are allowed only one choice: to do direct transmission of their copy to the destination node, if, in time,

they make direct contact with it (the "wait" phase). The n carrier nodes in the spray phase may be chosen to be the first distinct n nodes encountered. A better option is the *binary spray scheme*, in which the sender distributes roughly half of the message copies it holds (initially n) to any carrier node it encounters, with all carriers then also further spraying half of their copies; the spraying process ends at a node when there is a single remaining message, at which point the node switches to the wait phase.

MaxProp (Burgess et al., 2006) implements a contextual optimization to its buffer management, to increase the message delivery rate and decrease the latency of delivered messages. A node prioritizes the messages waiting to be forwarded based on a cost assigned to each destination node in the network. This cost is an estimate of delivery likelihood, and is updated continuously during the lifetime of the network. The estimation works by first approximating the likelihood that a node will next meet a certain other node; these estimations are exchanged between nodes at connection times, and effectively allow each node to approximate shortest paths to any destination in the network. A complementary mechanism uses end-to-end acknowledgments of packet deliveries to clear out local buffers.

2.2. Movement model in DTNs: random waypoint with shortest paths

Computational models of urban DTNs require realistic modelling of the free, stochastic movement of agents of different types (pedestrians, cars, boats, etc.). In our experimentation (see Section 5), we have used the Opportunistic Network Environment simulator The ONE (Keränen, Ott, & Kärkkäinen, 2009). In The ONE, agents following a specific movement model are first associated to a specific *map layer*, which describes the physical paths reachable by that type of agents. Then, a node moves as per a set of *points of interest (POIs)* located on that map layer. In the *random waypoint with shortest paths* movement model, a node randomly chooses a next destination point from a set of points of interest; the node travels there at a realistic speed on the shortest path possible on that map layer, and takes a break. Then, it repeats the process.

Therefore given a realistic map scenario, the combination of map layer, the set of points of interest, the range of speeds allowed, and the duration of the breaks will differentiate the movement model of a city car from that of a small boat and from that of a pedestrian. For example, the two urban scenarios tested in our experiments are configured with two map layers: one accessible to pedestrians, and one to motorized vehicles (either boats or cars).

2.3. Types of security attacks

Similar to a DTN routing protocol, an attack protocol may use a degree of network knowledge. A *weak attacker* cannot predict any future network conditions; his/her only resort consists of simply randomly attacking nodes in the network. A *strong attacker* is that with full network knowledge, who can predict the future pattern of network encounters: for example, a statistical estimation of how many honest nodes will be in the proximity of any given map location.

A group of *colluding attackers* is that which has the means necessary to synchronize and distribute their individual attacks, rather than execute an independent attack protocol on each node. Any of the colluding nodes may adopt one of the following basic attack logics, which we study here:

- **Black-hole attacks**: The attacker silently drops all the messages received, rather than store and forward them.
- **Flooding attacks**: The attacker executes the same routing protocol as the honest nodes in the network, but attempts a denial-of-service procedure by injecting a (large) number of (large) messages into the network.

Other known attack strategies include the falsification of routing tables to mislead network nodes (effective in the case when a routing protocol transmits routing metadata as part of replicated messages), counterfeiting acknowledgement messages, and the impersonation of other nodes (effective against protocols which build trust relationships between nodes).

Finally, the model of an attacker may be *free range*, i.e., the attackers can freely select their movement model (including the set of points of interest on the relevant map layer); this general model is studied here. On the contrary, an attacker may also have no freedom of movement. A scenario for this latter model is that in which the communication system of an honest network user is corrupted by the attacker, while the honest user still dictates the physical route; this scenario was studied in some of the related work (see the next section).

3. Related work

In this section we distinguish four categories of algorithms used for testing the robustness of DTNs against attacks, and briefly survey the state-of-the-art algorithms belonging to each category. Overall, the current related work studies either relatively small DTN scenarios, or heterogeneous attack methods among attackers with no freedom of movement. It never studies, though, the effectiveness of heterogeneous attackers with free range of mobility.

3.1. Random sampling

The conventional method to test the resilience of network protocols against attacks uses simulation and a random injection of events of interest. Such a computational study of DTN robustness was performed in Burgess et al. (2007), which evaluated the theoretical resilience of the real-world DieselNet and Haggle DTN prototypes based on long-term trace recordings of these networks' mobility and connectivity patterns. In particular, the DieselNet experiment consisted of roughly 30 sparsely connected regional buses communicating via WiFi. The connectivity traces used showed that Haggle nodes mimic the random-waypoint mobility model fairly closely. Four DTN routing protocols, variations of MaxProp (Burgess et al., 2006) (both forwarding and replicative), are evaluated post-factum over these prerecorded traces. Weak attacks were simulated by randomly reassigning some of the honest nodes as attackers (i.e., these attackers have no free range). A protocol similar to First Contact is then shown to only lower its data delivery ratio by 50% (under the baseline DDR obtained in the network without attackers) in the situation when 50% of the number of honest nodes in the network are corrupted by either black-hole or flooding attackers. The replicative MaxProp is shown to be more robust to weak attacks: with the same large number of attacks, the drop in data delivery was roughly 20% under the baseline.

Choo, Chan, and Chang (2010) further evaluated MaxProp on the same prerecorded traces, showing it to be less robust when a small number (one to five) of random attackers execute two attack logics concurrently: flooding and node impersonation. The drop in DDR with only five such combined attackers was found to be roughly 30% under the baseline without attacks.

This related work is less than general: it studies the resilience of DTN protocols on the same, relatively small and sparse DTN prototypes, post-factum, i.e., essentially constraining network nodes to one mobility pattern across all experiments. In the present work, we lift this constraint: we experiment with networks an order of magnitude larger (200 nodes), and study the resilience of the protocols under the general random-waypoint mobility model.

3.2. Greedy heuristics

A greedy heuristic was proven to compute approximations of the absolute optimum solutions to the problem of selecting the nodes with highest viral influence in static social networks (Kempe et al., 2015), given specific stochastic protocols dictating how a node may spread its influence in the network. This problem is related to ours, and was proved NP-hard in Kempe et al. (2015) by reducing it to the NP-complete vertex-cover problem. With some constraints upon the allowed rate of change in the network, the approximation algorithms are also practically interesting for solving similar open questions from the domain of dynamic social networks (Zhuang, Sun, Tang, Zhang, & Sun, 2013).

To generate strong attacks (again, without free range, and on DTNs of fixed, prerecorded movement models), Burgess et al. (2007) used a greedy heuristic to maximize the effect of black-hole attacks. As a first note, their method does not aim to minimize the delivery rate of the network, but substitutes it with the simpler quality-of-service metric of *total reachability*, defined as follows. A DTN model $D = (N, C)$ is a single *predefined list of connection events* C on the set N of n honest nodes; this is obtained simply from the prerecorded DTN trace used. Two nodes are temporally connected in D if there exists a (temporally non-decreasing) sequence of connection events in C. Then, the total reachability of D, denoted $R(D)$, is the number of pairs of temporally connected nodes. To select k attackers out of the set N while minimizing $R(D)$ on a predefined D, a *greedy heuristic* simply selects as the ith attacker that node which lowers the total reachability of D (excluding the first $i-1$ attackers) the most. While this greedy heuristic is not proven analytically to be optimal at minimizing $R(D)$, it is shown experimentally to give similar results as a brute-force method, for $k \leq 5$, on the two DTN traces under study.

The greedy heuristic in Burgess et al. (2007) is shown to outperform a random sampling of nodes when $k \geq \frac{n}{10}$ (i.e., at least 3 strong attackers for $n = 30$ honest nodes) on the DieselNet DTN, and for any $k \geq 1$ on the Haggle DTN, only for black-hole attacks and the MaxProp routing protocol. The heuristic is particularly advantageous for a very large k; there, the delivery rate when k equals $\frac{n}{2}$ is found to be under 20% (and reaches nearly zero for DieselNet).

This design of a greedy heuristic has the following limitations, which are then addressed in the present work: (1) a single attack method is studied, i.e., all attackers in the group perform the same attack logic; (2) the attackers studied are not allowed free range, but must follow the route of a previously honest node in that DTN; (3) the message delivery rate of a DTN D is evaluated via a proxy metric, the total reachability $R(D)$.

3.3. Evolutionary algorithms

Evolutionary algorithms (EAs) have been traditionally used as optimization tools for various applications. However, thorough observations of EA dynamics Auerbach, Iacca, and Floreano (2016); Pugh, Soros, Szerlip, and Stanley (2015) have recently suggested the idea that EAs work best as an *explorative* tool, rather than *exploitative* one (i.e., an optimizer). This intuition has originated new breeds of EAs which are based on mechanisms for preserving (or even promoting) the population diversity (Squillero & Tonda, 2016): rather than refining the search over a basin of attraction of the landscape, these algorithms foster the spread of new solutions over the search space in the attempt of finding uncovered, possibly *rare* instances of a given problem solution. Example of this kind of EAs are Novelty Search (Lehman & Stanley, 2011) and MAP-Elites (Mouret & Clune, 2015).

In the networking domain, the exploration capabilities of EAs have been used recently to devise stress-testing methods for network protocols, with the aim of maximizing or minimizing certain (functional or non-functional) network performance factors, such as message throughput or energy consumption. In this context, the key idea is that the working conditions which lead the network to a decreased performance are *anomalous* and, possibly, *rare* and hard to find (they are, essentially, a needle in the haystack). However, modern EAs can tackle this problem efficiently: in Bucur, Iacca, Squillero, and Tonda (2014), we tested this idea on stress-testing routing protocols for ad hoc Wireless Sensor Networks (on this topic, see also: Alcaraz and Lopez (2010); Brooks, Pillai, Racunas, and Rai (2007); Islam, Shen, and Wang (2012); Rogers, David, and Jennings (2005)). We then extended the applicability of this methodology to DTNs in Bucur et al. (2015), a feasibility study of the present work; there, we used an evolutionary algorithm to determine small groups of *homogeneous* attackers (i.e., nodes performing the same kind of attack) capable of compromising a DTN. In Bucur et al. (2016), we instead tackled this problem from a co-operative co-evolution perspective, which led to a further decrease of the network performance.

3.4. Other approaches

Recently, alternative approaches have also been considered to assess the security of mobile networks. In Li, Yang, and Wu (2010), a game-theory-based framework is proposed to model and analyse the behaviour of a group of 40 malicious nodes attacking a mobile ad hoc network made up of 100 nodes in total. Malicious nodes can perform attacks of two types: dropping packets (similar to a black-hole attack, but the choice of dropping a packet is made for each packet), or altered packets (in this case the packet is forwarded after a malicious modification). Honest and malicious nodes are modelled as rational decision-making agents aiming to maximize their own individual utility function. A limitation of this framework is that, while honest nodes are allowed to cooperate in order to alleviate the effect of the attack, no form of collusion among malicious nodes is allowed. In the present work, we instead consider colluding attackers, which may "distribute" the work and thus more effectively point to weaknesses of the protocol under study.

4. Proposed methods

In this section we describe the two proposed search algorithms for attack-effect maximization. Both algorithms receive in input a DTN scenario consisting of:

- A set of (possibly overlapping) **map layers** L (e.g., city streets or pedestrian walkways), each defined as a set of line segments between pairs of map points from a set. The set of map points for a given map layer l is denoted l.POIs;
- A set of **movement models** M for nodes, each corresponding to a type of mobile agent in the scenario (e.g., pedestrians or cars) and each constrained to a single map layer from L: $\forall m \in M, \exists l \in L : m.\text{map} = l$;
- A preset **network size** n and a distribution of the n nodes among the movement models M, $n = \sum_{i=1}^{|M|} n_i$ (e.g., a DTN scenario may have $n = 200$ and consist of $n_1 = 150$ pedestrians and $n_2 = 50$ cars);
- A set of possible **attack logics** to choose from, A (e.g., black-hole or flooding), kept constant for each node during the attack;
- An interval $[k_{\min}, k_{\max}]$ constraining the colluding attackers' **group size**, with $k_{\min} \geq 1$ and $k_{\max} \leq n$.

The algorithms will then output a maximal-effect strong attack configuration; this is a valuation of the following parameters:

- A concrete number of colluding attackers $k \in [k_{\min}, k_{\max}]$;

Algorithm 1 Greedy heuristic to compute exactly $k_{\max}$ attack tuples for each possible combination of attack logic and movement model.

procedure GREEDY($p, L, M, A, n = \sum_{i=1}^{|M|} n_i, k_{\max}$)
  $D \leftarrow$ the DTN without attackers $L, M, n = \sum_{i=1}^{|M|} n_i$
  Sample 10 random evaluations of D
  for each $m_i \in M$ **do**
    $P_i \leftarrow p$ most visited map points in m_i.map.POIs
  end for
  $k \leftarrow k_{\max}$
  for each tuple $(a_j, m_i) \in A \times M$ **do**
    $K_{ji} \leftarrow k$ times the tuple (a_j, m_i, P_i)
  end for
  Return all sets K_{ji}, $1 \le j \le |A|$ and $1 \le i \le |M|$
end procedure

- A set K of exactly $|K| = k$ **attack tuples**

  $(a_l \in A, m_l \in M, P_l \subseteq m_l.\text{map.POIs})$ where $1 \le l \le k$.

In essence, every malicious node l is free to select its attack logic a_l, movement model m_l (and thus map layer), and a subset of map points P_l from this map layer; this set dictates the route of the attacker, which will follow the same model (random waypoint with shortest paths) as the honest nodes, except for the fact that the set of waypoints is now strictly P_l.

In our study, the k attackers added to the network will replace k honest nodes, so that the network size remains equal to n, and a simple calculation can be made on the proportion of malicious nodes out of n. This entails that, when generating k attack tuples, we have a further constraint in that the number of attackers which can be assigned to a certain map layer $m_i \in M$ is upper-bounded by the number of honest nodes on that map layer. In practice though, our network size n is one order of magnitude larger than the largest $k_{\max}$ used, so that this constraint is always satisfied.

4.1. Greedy heuristic

We use a computationally efficient, greedy procedure to set k and calculate the set K of k attack tuples (a_l, m_l, P_l), with $1 \le l \le k$ and $k \in [k_{\min}, k_{\max}]$. The procedure (shown in Algorithm 1) uses the DTN configuration $L, M, A, n = \sum_{i=1}^{|M|} n_i$, and a supplementary, configurable parameter p, which is the size desired for the P_l of any attacker l.

Intuitively, this heuristic achieves the following: it first evaluates the DTN without attackers; after this, it calculates, for every map layer, the set of points most visited by honest nodes, selects a number of the top such map points, and assigns them to all attackers mobile on that map layer. Essentially, this heuristic attempts to optimize the number of connection events in a general way, by obtaining the "heat map" for our DTN geographical locations, in a way suitable to a DTN for which we do not have, as (Burgess et al., 2007) did, prerecorded logs of connection events.

Our heuristic could also be called a *dynamic* version of a greedy heuristic used for computing the nodes of maximum influence in static social networks (Kempe et al., 2015): given a fixed directed graph, that heuristic selected a subset of nodes which had the highest degree in the graph. Here, it is the geographic location of nodes which enable communication, so our heuristic considers the heat map of the most visited geographic points to be the graph underlying the DTN, and greedily selects a subset of nodes which have the highest number of visits.

The heuristic outputs $|A| \times |M|$ sets of attacker groups: rather than trying to maximize the effectiveness of a single set K, the heuristic computes a set K_{ji} for each combination of attack logic $a_j \in A$ and movement model $m_i \in M$. With this, the procedure essentially tries to "place" a single type of attack on each map layer, leaving to a later step the task of evaluating which among these combinations is best. Then, it maximizes the number of attackers in each set K to exactly $k = k_{\max}$. Finally, it greedily computes a single set of map points P_i for all the attackers of movement model m_i; this is the set of the p most visited map points on m_i.map, estimated from repeatedly sampling the execution of the network without attackers. Note that assigning the same set of POIs P_i to all attackers in the group is arguably very beneficial, because a single attacker can only be at one such map point at one time; on the other hand, when another attacker also travels to the same map points, these points will have better coverage.

4.2. Evolutionary algorithm

The evolutionary algorithm is applied here to maximize the effect of the attackers on the network, by optimizing the attacker team composition (i.e., number and types of attackers) and movement. A candidate solution — or *individual* in the evolutionary jargon — represents in this case a set K (of variable size) of attack tuples, defined as above.

Two network metrics are used as *fitness* functions: the data delivery rate, calculated as the percentage of messages originated *only* from honest nodes, and which are delivered successfully (to be minimized); and the average latency of message deliveries, in seconds (to be maximized). These two metrics are considered in lexicographic order, so that the latency is considered as a tie-breaker in the rare case the evolutionary algorithm compares two different attacks with equal DDR.

We should note here that our purpose is not to implement an ad hoc evolutionary algorithm which may be optimally designed for our problem (or, to find the best EA for this problem). On the contrary, we rather ask the question whether (and how) a general-purpose EA, with no specific domain knowledge and parameter tuning, can be applied to this problem, and how it compares against a well-crafted problem-specific heuristic. For this reason, among the broad family of EAs, we chose an algorithm that was shown to perform fairly well on a number of real-world applications, namely the self-adaptive $(\mu + \lambda)$-Evolution Strategy (ES) (Hansen, Arnold, & Auger, 2015). In particular, we use here the implementation available in the evolutionary toolkit μGP[1] (Sanchez, Schillaci, & Squillero, 2011).

The ES proceeds as follows (Algorithm 2): first, a population of μ individuals is randomly initialized and evaluated. At each generation, λ new individuals are obtained by applying mutation and crossover with self-adaptive parameters: the first operator performs mutations at node level (i.e. changes of attack logic, movement model, or insertion/removal/replacement of POIs); the latter performs crossover between two individuals, i.e. swaps of attack types, movement models, or subsets of POIs between them. Then, offspring are evaluated and compete with parents, by tournament selection of self-adaptive size τ. This loop goes on until a stop criterion is met, defined in our case as a stagnation condition of § generations without improvement.

The number of parent and offspring individuals (μ and λ) are typically scaled with the problem dimension. However, this is not directly applicable to our problem, as we have variable-sized individuals (since the size k of the attack tuples set will change from an individual to another). Since here the maximum value of k was 20, and since excessively large populations would be slower

[1] http://ugp3.sourceforge.net.

Algorithm 2 Evolutionary algorithm to optimize the attacker team composition and movement. EvalDTN(D, K_i) simulates the DTN D attacked by the group of attack tuples K_i and returns the resulting DDR and latency. $min()$ and $max()$ are evaluated on all the attacker groups generated by the algorithm.

procedure EA($L, M, A, n = \sum_{i=1}^{|M|} n_i, k_{\min}, k_{\max}$)
  $D \leftarrow$ the DTN without attackers $L, M, n = \sum_{i=1}^{|M|} n_i$
  for $i \in [1, \mu]$ **do**
    Randomly initialize a set K_i of attack tuples, $|K_i| = k, k \in [k_{\min}, k_{\max}]$
    $(DDR_i, latency_i) \leftarrow$ EvalDTN(D, K_i)
  end for
  while stagnation condition is not met **do**
    Generate λ attacker groups by mutation and crossover
    for $i \in [1, \lambda]$ **do**
      $(DDR_i, latency_i) \leftarrow$ EvalDTN(D, K_i)
    end for
    Select μ attacker groups by tournament selection on $\mu + \lambda$ attacker groups
  end while
  Return K^* s.t. $(DDR^*, latency^*) = (min(DDR), \max(latency))$
end procedure

Table 1
DTN routing protocols under test.

Protocol	Configuration
First Contact	messages not replicated
Epidemic	buffer with drop-oldest-messages purge scheme
Spray and Wait	binary spray scheme; 5 copies per message
MaxProp	optimized buffer management with shortest path estimation and acknowledgements

to converge, we use $\mu = 30$ and $\lambda = 20$ as a compromise between exploration and exploitation in our problem space.

As for the tournament size τ, the algorithm available in μGP is able to self-adapt its value; we chose here the range ([1, 4]), i.e., we allowed the algorithm to adjust dynamically the selection pressure, by changing τ. For $\tau = 1$ the selection pressure is lower, as all individuals can reproduce; for $\tau = 4$, the selection pressure is higher, since each individual competes with three other randomly chosen individuals, and thus low-fitness individuals have less chances of reproducing.

Finally, the number of stagnated generations § was set to 50: this value is sufficiently large to allow the algorithm to continue the search while no improvements are found, but not too large to avoid wasting computational resources.

5. Experimental settings

This section presents the experimental configurations for the three routing protocols under study, the movement models in the DTN as determined by two urban scenarios, the communication settings in the network, which determine node connectivity patterns, and the settings for the evaluation of a DTN via simulation. We also maintain a repository.[2]

The settings of the four routing protocols (First Contact, Epidemic, Spray and Wait, and MaxProp, previously described in Section 2.1) are briefly summarized in Table 1.

[2] https://github.com/doinab/DTN-security.

5.1. Two urban scenarios

We simulate two realistic, large-scale city environments. Fig. 1 shows the basic maps of the two scenarios: two areas of San Francisco and Venice, each map composed of two map layers for pedestrians and motorized vehicles (cars in the first scenario and boats in the second). Each line segment is defined by two map points, and every intersection of line segments consists of a point. In Fig. 1 (left), the two map layers are largely overlapping, such that the pedestrian layer (in green, grey in print version) is mostly obscured by the street layer (in black). In Fig. 1 (right), the map layers do not overlap, with the exception of a small number of common map points located at bridges.

These cities differ in terms of map morphology: while the area of San Francisco has a regular grid structure of paths for both motorized vehicles and pedestrians, with only the occasional pedestrian-only park, the core of Venice has a complex, hierarchical, irregular morphology of main and secondary waterways travelled by boats, with pedestrians confined to inner walkways (some along waterways) and bridges (note that here we do not model, for simplicity, the scenario in which a vehicle carries pedestrian passengers, and thus has multiple communication devices active). The Venice map has an additional feature for added realism: on both map layers, a small number of the map POIs are special: they mark the touristic centre, and have a higher probability to be chosen as the next destination by the honest nodes (30% and 20% for pedestrians and vehicles, respectively).

In both cities we configured $n = 200$ mobile agents, divided in two types: pedestrians (75%) and vehicles. For San Francisco, the vehicles consist of motorized cars; in Venice, the waterways serve as routes for motorized or unmotorized boats. For an honest node, the set of POIs is the entire set of map points located on the node's relevant map layer. For an attacker, the set of POIs is computed either by the greedy heuristic or by the evolutionary algorithm. We set the parameter p given to the greedy heuristic to the value $p = 100$, i.e., the heuristic calculates the 100 most visited map points and sets these points as the set of POIs for all attackers. On the other hand, the evolutionary algorithm will yield a variable number of attacker POIs, limited only by the number of map points on the map layers in the scenario.

Table 2 quantifies the maps and map layers in terms of size, number of distinct map points, line segments, and number of nodes in the DTN scenario. The configuration for the nodes' movement models is given in Table 3.

5.2. Communication and attack settings

Pedestrians are modelled to carry communication devices with limited capabilities: a Bluetooth communication interface with a range of 15 m and low bandwidth. Vehicles have more communication capabilities: besides a Bluetooth interface (which allows communication events to take place between any pedestrian and any vehicle), a vehicle also has a high-speed, longer-range network interface allowing vehicle-to-vehicle communication. The simulation and communication settings are summarized in Table 4.

A number k of the 200 nodes is assigned malicious behaviour: an attacker executes either a black-hole attack or flooding, the two attack logics described in Section 2.3. We experiment with values of k between 1 and 20, i.e., up to 10% of the network size, in the following increasing intervals $[k_{\min}, k_{\max}]$: [1, 1], [2, 2], [1, 5], [6, 10], and [11, 20]. The intervals are relevant for the experimentation using the evolutionary algorithm, which will evolve k within one such interval. This serves to show whether or not, in some DTN scenarios, it is strictly necessary to maximize the attacker group size, or, on the contrary, whether a smaller k might not have a similar effect upon the network as a larger k.

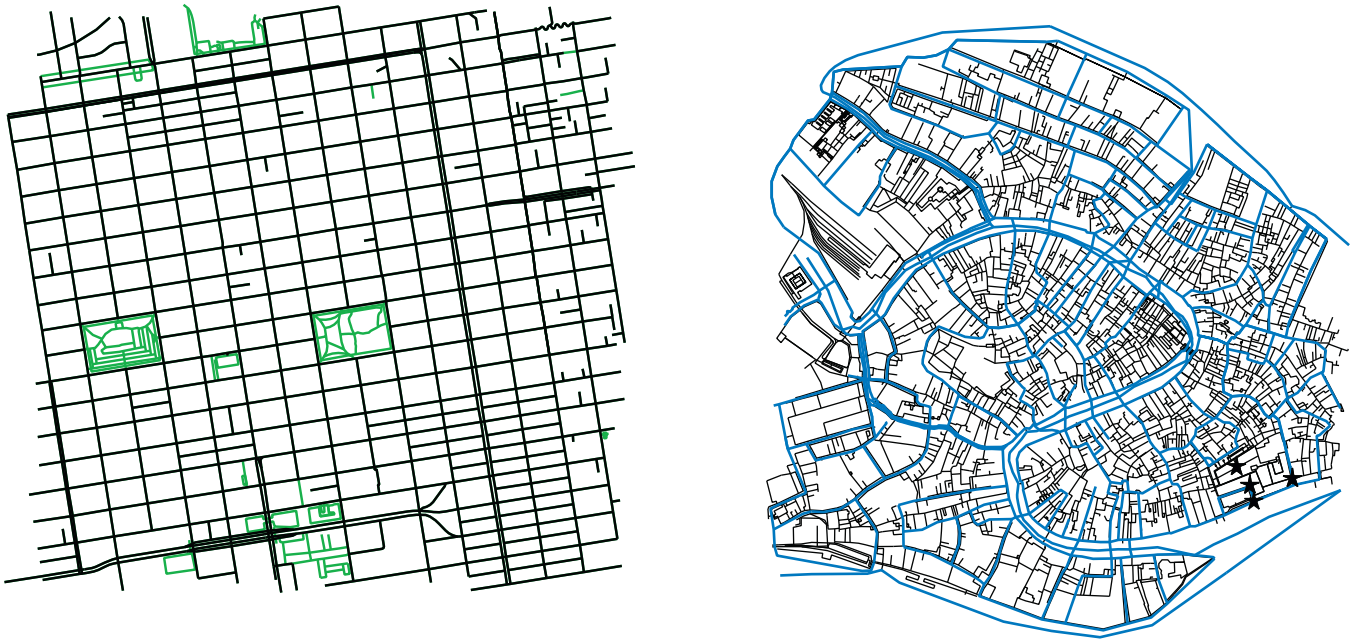

Fig. 1. (left) A 5 km^2 area of downtown San Francisco, US, with a grid-based map topology of streets and the occasional park. The map has two overlapping layers, constraining the movement of vehicles (the black map layer) and pedestrians (the green map layer, grey in print version). (right) A 5 km^2 area of downtown Venice, Italy, with an irregular map topology of pedestrian pathways (the black map layer) and waterways (the blue layer, grey in print version). Marked with stars are special POIs in the city's touristic centre. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Network parameters: city maps.

San Francisco:	size:	2416 m × 2253 m
	map layers:	L_P (pedestrian walkways), L_S (streets)
	no. of route segments:	1728 in L_P, 1305 in L_S
	no. of map points:	1210 in L_P, 883 in L_S
	network size n:	$n_1 = 150$ pedestrians (constrained to L_P), $n_2 = 50$ cars (constrained to L_S)
Venice:	size:	2210 m × 2340 m
	map layers:	L_P (pedestrian walkways), L_W (waterways)
	no. of line segments:	7983 in L_P, 1497 in L_W
	no. of map points:	6910 in L_P, 1354 in L_W
	network size:	$n_1 = 150$ pedestrians (constrained to L_P), $n_2 = 50$ boats (constrained to L_W)

Table 3
Network parameters: movement models.

Movement model for nodes in both cities	next point:	chosen randomly from a set of POIs
	path choice:	shortest path on the map layer to the next point
	pedestrian speed:	[0.5 . . . 1.5] m/s
	boat speed:	[1.0 . . . 5.0] m/s
	car speed:	[2.7 . . . 13.9] m/s
	pause interval for all:	[0 . . . 120] s at each destination point

Honest nodes periodically inject new messages to be routed by the network; the rate of message injection among all honest nodes is set at one message every 30 s, such that the network routes 120 honest messages per hour, regardless of the number of attackers in the network. The next honest node to inject the next message in the network is chosen randomly, as is the destination node for the message.

A black-hole attacker does not inject any additional messages in the network. On the other hand, when an attacker executes a flood, the parameters are chosen to obtain a "heavy" flood of messages: (1) a flooding node injects messages in the network at 10 times the frequency of message injection from an honest node, and (2) the messages injected by a flooder are 10 times as large as regular messages. Table 4 also summarizes these communication parameters, together with the settings regarding the sizes of the nodes' message buffers, and the Time To Leave (TTL) for all messages.

5.3. Experimental campaigns

Each simulation of a DTN in The ONE simulator is stochastic (Keränen et al., 2009). The nodes are initially placed randomly on their map layer, and a 1000-s warm-up simulation period is allowed before the experiment starts, for the nodes to settle on the emerging pattern of preferred routes in the city. Every next waypoint is also chosen randomly from the relevant set of map points, as per the general random-waypoint with shortest paths node movement model.

Due to this, we evaluate each network scenario via *10 simulation repetitions* with different random seeds, and report the *average* data delivery ratio for the network (and message latency, if relevant). This number of repetitions was chosen as a trade-off between computational costs and the width of the 95% confidence interval around the mean; as will be seen in Section 6, this width of the confidence interval with 10 repetitions is low,

Table 4
Network parameters: simulation and node communication settings.

Simulation settings	simulation time:	5 h
	DTN simulator:	The ONE (Keränen et al., 2009)
Message settings	message issued:	every 30 s (by an honest node), every 3 s (by a flooder)
	message size:	10 kB (issued by an honest node), 100 kB (issued by a flooder)
	message buffer:	5 MB (for pedestrian nodes), 50 MB (for car and boat nodes)
	message TTL:	5 h
Node communication interfaces	Bluetooth:	range 15 m, speed 250 kBps
	High-speed:	range 100 m, speed 10 MBps
	pedestrians use:	Bluetooth
	cars and boats use:	Bluetooth and High-speed

on the order of 1% of the mean. This number of simulation repetitions is used uniformly for all computational methods in this study, i.e., when evaluating *any* given DTN D (either without attackers, or attacked by the group of attack tuples K). Thus, a greedy experiment (Algorithm 1 described in Section 4.1) will evaluate the network configuration D without attackers via 10 simulation repetitions, and use the aggregated simulation logs to compute the p most visited map points. Also, any evolutionary experiment (Algorithm 2, Section 4.2) implements the function which evaluates a DTN, EvalDTN(D, K), via 10 simulations.

Finally, each of the three DTN protocols under study (Table 1) is executed in each of the two urban scenarios (defining the map layers L, node movement models M, and fixed network size $n = 200$), while setting the number k of strong attackers to within each of the intervals in the set {[1, 1], [2, 2], [1, 5], [6, 10], [11, 20]}, and fixing the set of attack logics A to black-hole and flooding attacks. For each combination of a DTN protocol with an urban scenario and an interval of attacker group size, attacks are computed in the following four experimental campaigns:

- **(no attack)** As a baseline for comparison of DTN resilience to attacks, we first evaluate the DTN D in which all the nodes are honest.
- **(random sampling)** As a baseline for comparison among attack-computation algorithms, we also evaluate the effect of 150 randomly generated sets of attack tuples K, where:
  - for a fair comparison, $|K|$ is set to $k_{\max}$, and
  - any attack tuple from K uses a valid map layer and movement model from the DTN scenario.
- **(greedy heuristic)** Algorithm 1, with p set to 100, outputs 4 sets of attacker tuples, one per combination from the set $A \times M$, as all scenarios have exactly two possible attack logics and two map layers and movement models ({black-hole, flooder}$\times$ {pedestrian, vehicle}). We evaluate all four attacks, which we refer to as: **BP, BV, FP, FV**, i.e., respectively, *black-hole pedestrian, black-hole vehicle, flooder pedestrian*, and *flooder vehicle*.
- **(evolutionary algorithm)** Algorithm 2 is run 5 times, with a different initial random seed. We report the best attack found among all runs.

6. Analysis of results

This section presents the numerical results of maximizing the effectiveness of malicious behaviour over the four DTN protocols and two maps, comparatively for the greedy heuristic, the evolutionary algorithm, the random attacks, and the baseline performance of the network without attacks. We discuss the fact that we observed great differences in the resilience of these protocols to strong attacks, and also in the type of strong attackers which are most effective over these protocols. We see that the effect of the underlying map (and thus, urban scenario) upon the effectiveness of an attack is minor in comparison with the attack logic and attacker's route. We see that both the greedy heuristic and the evolutionary algorithm have their own strengths when used to compute solutions for this problem. Finally, we quantify the runtime of the evolutionary algorithm; the runtime of the greedy heuristic is relatively negligible.

6.1. Protocol performance: First Contact

In Fig. 2 (Left) we compare the attack effectiveness on the First Contact protocol using the four experimental campaigns described in Section 5.3. Out of the four greedy heuristics, we present only the most successful among them: that which generated groups of vehicle attackers executing a black-hole attack logic (BV); in the two urban scenarios, these greedy vehicle attackers are cars and boats, respectively. We draw the following conclusions:

Protocol resilience. While First Contact has low data delivery even without injecting attacks into the network (a delivery ratio under 40% in both urban scenarios), our algorithms could produce single strong-attack strategies which lowered data delivery under 20%, and also multi-attacker strategies which further lowered DDR under 5%. This shows that there exist strong attacks which overcome the routing logic of First Contact. While non-replicative protocols are known to be weaker than replicative ones, we show the extent of this weakness better than the related work, as follows. In (Burgess et al. 2007) it was found that a non-replicative protocol lowers its data delivery ratio by 50% (under the baseline without attacks) only when 50% of the number of honest nodes in the network are corrupted by either black-hole or flooding attackers (with the important note that these attackers do not have free range over the urban map). Here, we see that giving the attackers free range achieves the same 50% drop in performance, but already by introducing in the network a *single* attacker rather than a large number.

Attack group composition. We look at which attack group composition is most effective. In what regards the greedy heuristic, Fig. 2 already shows that it is attackers of the black-hole vehicle (BV) type which are single-handedly most effective. However, by construction the greedy heuristic only tests homogeneous groups of attackers. Instead, the evolutionary algorithm is free to independently optimize the type of each attacker in the group, as well as the group size; we hypothesize that this optimization strategy can be more effective than the greedy heuristic, particularly in urban scenarios where the city map is irregular. We thus look at all the attack tuples generated in the process of repeatedly executing Algorithm 2; we then select all the attack tuples which lower the network's data delivery to a value within 2% of the best one obtained overall, and perform statistics on this sample set of attack tuples. The sample size obtained is always on the order of magnitude of 10^2 to 10^3 attack tuples, across all scenarios.

In Fig. 2 (Right), we show the average attacker group size and composition among the top groups, for First Contact, as obtained by the evolutionary algorithm. The algorithm found that a group can be equally as effective even when its size is not maximized: for 6 to 10 attackers in the group, the average size of a top group is 9.18 in the San Francisco scenario, from a sample of nearly 8000 top groups. The majority of attackers in all top groups are black-

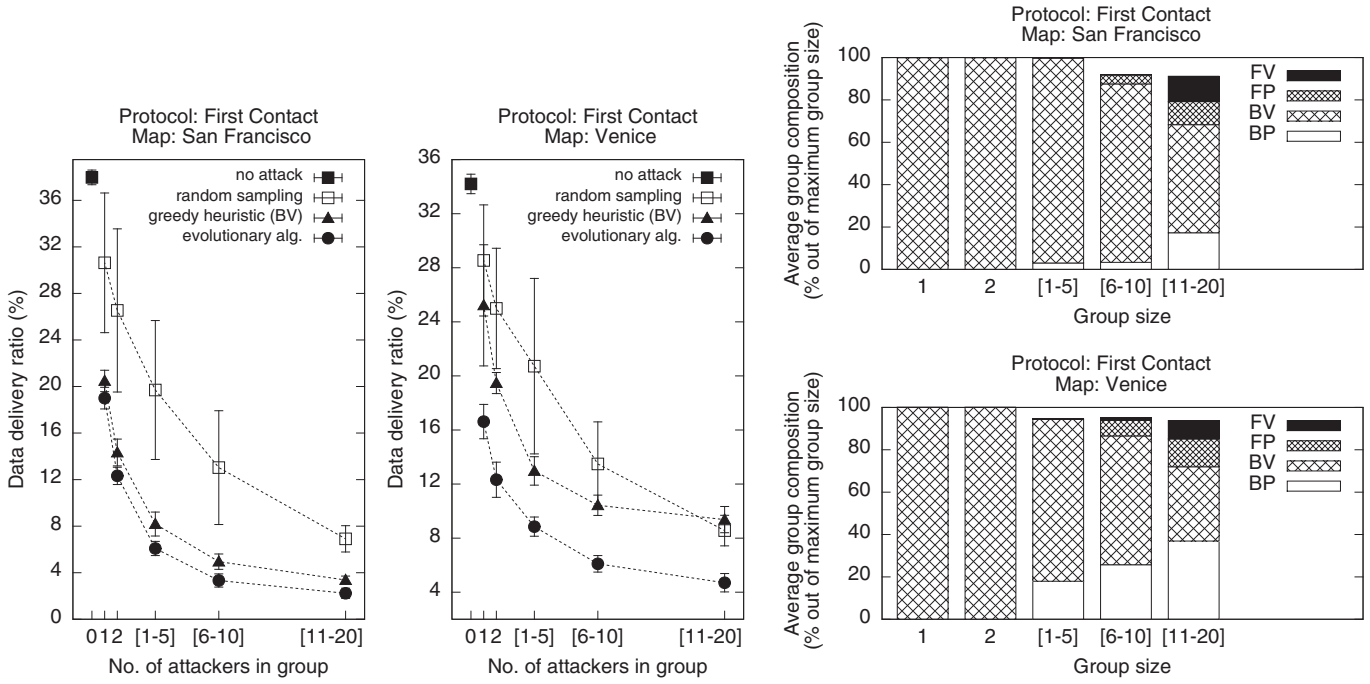

Fig. 2. (Left) Attack effectiveness on First Contact: data delivery ratio (DDR) with (1) no attack, (2) randomly computed attacks, the best attacks computed by (3) the greedy heuristic and (4) the evolutionary algorithm. Cases (1), (3) and (4) reflect a single DTN, and are shown as the DDR mean and 95% confidence interval; case (2) is shown via the DDR mean and standard deviation among 150 DTNs. "BV" denotes the greedy heuristic which generates groups of *black-hole vehicle* attackers, in both scenarios. **(Right)** The average composition of the top attacker groups found by the evolutionary algorithm for First Contact.

hole vehicles in the San Francisco scenario; for Venice, the group composition is a more balanced mix, with a fair percentage of black-hole pedestrians, which reflects the nature of that city map. Overall, *black-hole attacks* lower the data delivery of First Contact most effectively, and faster (i.e., vehicle rather than pedestrian) black-hole attackers are advantageous.

Although the city maps have different route morphology, Fig. 2 shows that attacks could be computed such that the data delivery was lowered to roughly the same extent in the two scenarios. We give an example of attack route in Fig. 3, which presents the POIs of the top single attackers for First Contact. The attacker's route consists of all shortest paths between any pair of POIs from the set (as per the movement model in Section 2.2). The attack on the grid-like map shows that the attacker covers the city streets uniformly; on the Venice map, the attacker navigates among only four POIs, travelling only via those canals which naturally form the backbone of the city.

6.2. Protocol performance: Epidemic

In Fig. 4 (Left) we compare the attack effectiveness on the Epidemic protocol using the four experimental campaigns. Out of the four greedy heuristics, we present the best two, which are relatively equally successful over this protocol: they both generated groups of attackers executing a flooding attack logic.

Protocol resilience. Without attackers, the Epidemic routing protocol maintains a data delivery ratio over 90% in both scenarios. Replicative protocols are expected to also be resilient to attacks (see Section 3): a number of attackers at least equal to 50% of the network size is needed to lower the DDR to under 20%, in the related work studying attackers without free range (Burgess et al., 2007). Our heuristics found that a *single*, free-range, strong attacker can already lower the network delivery rate to 30% regardless of the city map, and also that, counter-intuitively, teams of attackers are not significantly more advantageous over the Epidemic protocol than a single attacker.

Attack group composition. Two of the greedy heuristics are more successful than the evolutionary algorithm, on both maps (as shown in Fig. 4 (Left)). In both scenarios, the greedy heuristics generate homogeneous groups of flooding attackers, of which the most successful attacks consist of flooding vehicles; similarly, the average group composition among the top groups found by the evolutionary algorithm is largely made up by flooding vehicles (Fig. 4 (Right)), especially for small attacker group sizes.

6.3. Protocol performance: Spray and Wait

In Fig. 5 (Left) we compare the attack effectiveness on the Spray and Wait protocol using the four experimental campaigns. Out of the four greedy heuristics, we present the best two per scenario, which either outdo or have relatively similar performance to the evolutionary algorithm. For Spray and Wait, the two best greedy heuristics differ between the two scenarios.

Protocol resilience. Spray and Wait, configured with the more sophisticated binary spray scheme, is a relatively smart and economical message-replication method. Among the DTN routing protocols studied here, Spray and Wait was the second-most difficult to optimize attacks for: with up to 20 attackers in the group, our algorithms were able to optimize attacker groups so that the data delivery ratio dropped by up to 20% under the baseline without attacks (as shown in Fig. 5 (Left)). However, unlike the cases of both First Contact and Epidemic, the drop in delivery ratio does not taper off with an increasing number of attackers, which gives that a larger attack group is significantly more advantageous over this protocol.

Attack group composition. Also unlike the other protocols we study, none of our optimization algorithms outdid the others across all the experimental settings. Instead, for Spray and Wait, the evolutionary algorithm and at least one greedy heuristic shared the best attack optimizations across group sizes. The homogeneous groups obtained by the greedy heuristics are advantageous for small group sizes; these are groups made up by *flood-*

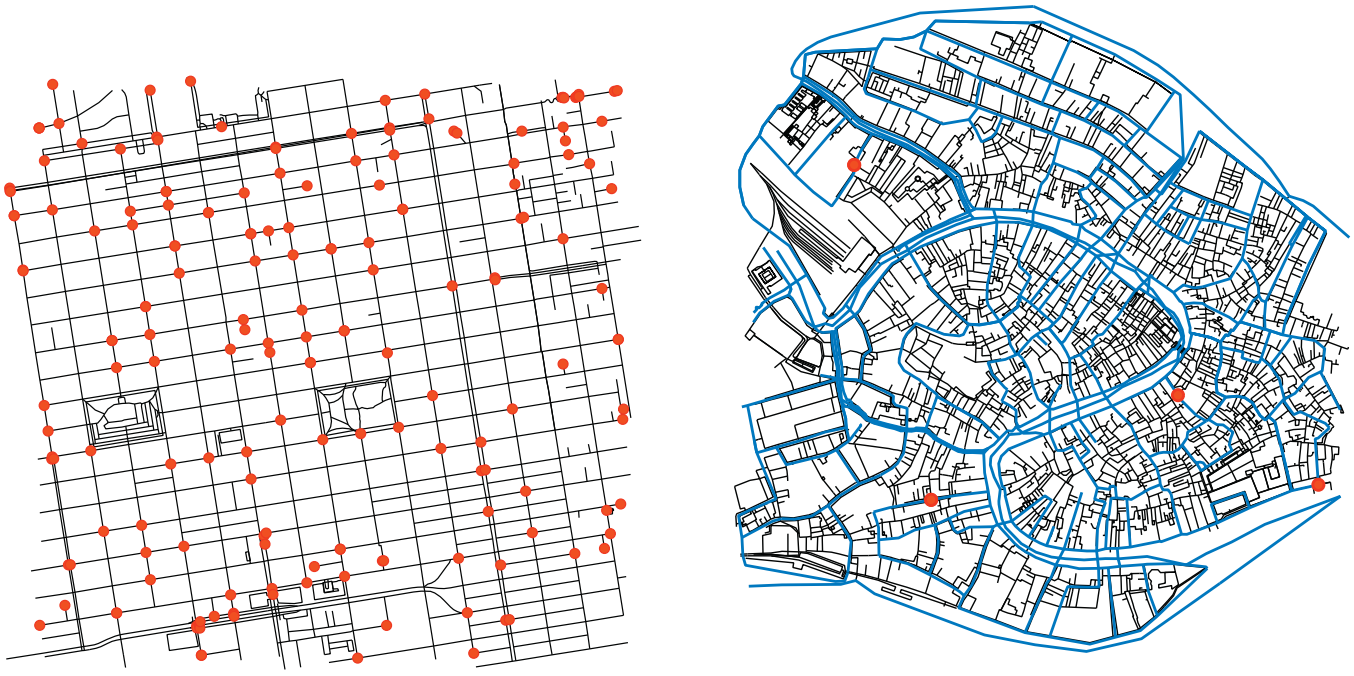

Fig. 3. The set of POIs (red dots, grey in print version) of the best single attacker in the San Francisco (left) and Venice scenarios, for First Contact. Both attackers are black-hole vehicles (a car and boat, respectively). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

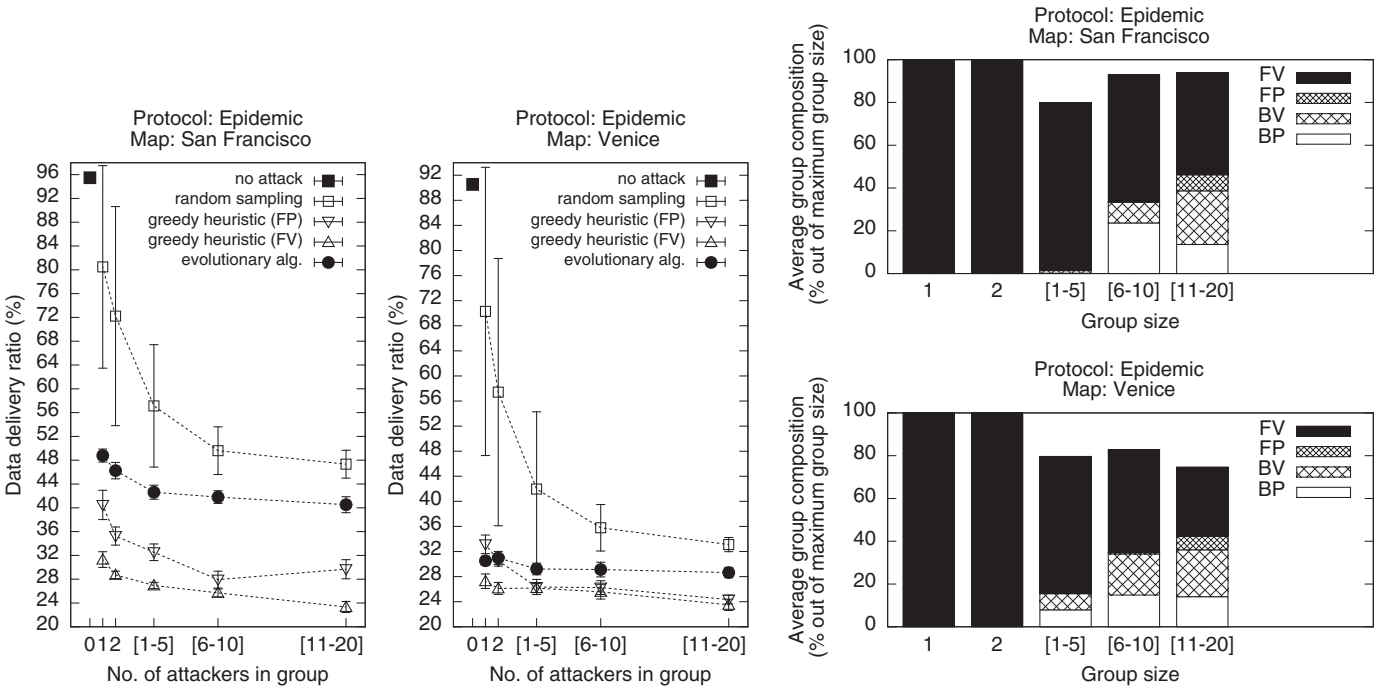

Fig. 4. **(Left)** Attack effectiveness on Epidemic: data delivery ratio (DDR) with (1) no attack, (2) randomly computed attacks, the best attacks computed by (3) the greedy heuristic and (4) the evolutionary algorithm. Cases (1), (3) and (4) reflect a single DTN, and are shown as the DDR mean and 95% confidence interval; case (2) is shown via the DDR mean and standard deviation among 150 DTNs. FP denotes the greedy heuristic which generates groups of flooding pedestrian attackers, and FV that of flooding vehicle attackers. **(Right)** The average composition of the top attacker groups found by the evolutionary algorithm for Epidemic.

ing attackers, either vehicles or pedestrians. On the other hand, the mixed groups obtained by the evolutionary algorithm for larger group sizes are more efficient, and the group composition among the top attack groups shows a mix of black-hole vehicles and flooding pedestrians (Fig. 5 (Right)).

6.4. Protocol performance: MaxProp

In Fig. 6, we compare the attack effectiveness on the MaxProp protocol using the four experimental campaigns. Out of the four greedy heuristics, we present the best two per scenario, both employing a flooding attack logic.

Protocol resilience. MaxProp is the most resilient of this set of four protocols: for both maps, the drop in performance by introducing up to 20 attackers in the network is approximately 8% under the baseline without attacks, compared to approximately 20% under the baseline for Spray and Wait, and 70% under the baseline for Epidemic. Similarly to the case of Spray and Wait, the drop under the baseline increases slowly and linearly with the size of the

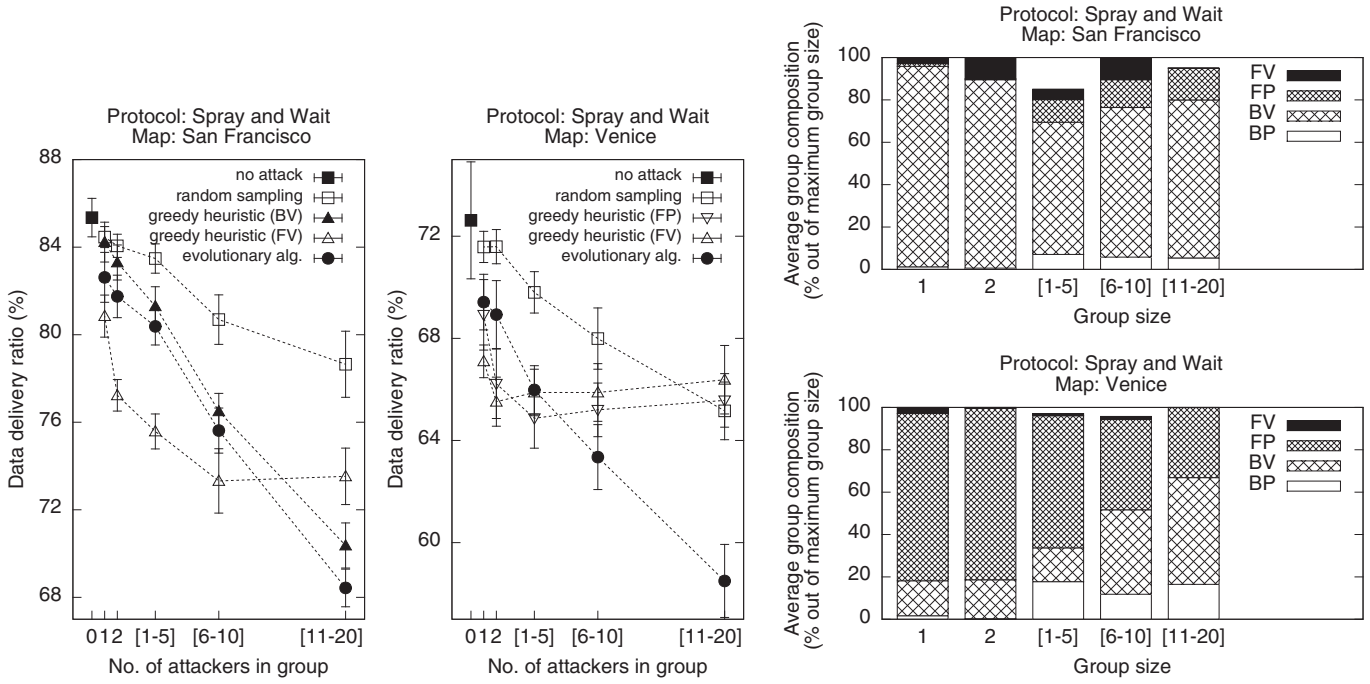

Fig. 5. (Left) Attack effectiveness on Spray and Wait: data delivery ratio (DDR) with (1) no attack, (2) randomly computed attacks, the best attacks computed by (3) the greedy heuristic and (4) the evolutionary algorithm. Cases (1), (3) and (4) reflect a single DTN, and are shown as the DDR mean and 95% confidence interval; case (2) is shown via the DDR mean and standard deviation among 150 DTNs. BV denotes the greedy heuristic which generates groups of black-hole vehicle attackers, FV that of flooding vehicle attackers, and FP that of flooding pedestrian attackers. **(Right)** The average composition of the top attacker groups found by the evolutionary algorithm for Spray and Wait.

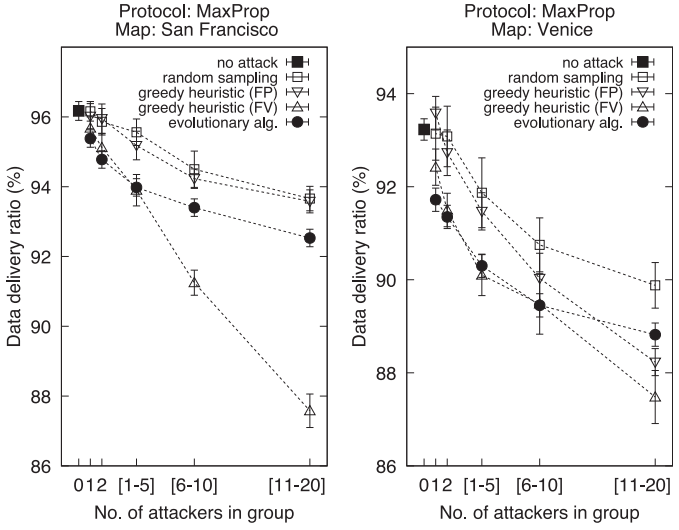

Fig. 6. Attack effectiveness on MaxProp: data delivery ratio (DDR) with (1) no attack, (2) randomly computed attacks, the best attacks computed by (3) the greedy heuristic and (4) the evolutionary algorithm. Cases (1), (3) and (4) reflect a single DTN, and are shown as the DDR mean and 95% confidence interval; case (2) is shown via the DDR mean and standard deviation among 150 DTNs. FV denotes the greedy heuristic which generates groups of flooding vehicle attackers, and FP that of flooding pedestrian attackers.

attack group, which gives that adding further attackers may continue to affect the ratio of delivered messages proportionally.

Attack group composition. As is the case for the simpler Epidemic protocol, fast flooding attackers are single-handedly successful, and, on the grid map of San Francisco, by a relatively large margin compared to the next best single attack logic. Thus, for this protocol, we found that the composition of the attack groups is unusually homogeneous, with no black-hole attackers in the top groups, and uniformly only flooding vehicles in the most effective large attacker groups over the grid map. As for the previous protocols, in the more morphologically complex Venice map with better focus on pedestrian-only routes, the effectiveness of pedestrian flood attacks is relatively close to that of vehicle flood attacks.

6.5. Runtimes of the experimentation

We conclude this section with a brief analysis of the runtimes of the evolutionary algorithm. A summary of the average number of core-hours and the number of generations until convergence ($\S = 50$ generations without improvement) is reported in Fig. 7, for each experiment (protocol/city/attacker group size).

The main observation here is that, with a few exceptions, in all cases the average number of core-hours to reach the stagnation condition increases (almost linearly) with the attacker group size. This increase can be intuitively explained because the size of the search space (i.e. the number of parameters describing the attacker group) increases with the maximum number of attackers allowed to the algorithm, thus requiring a larger number of evaluations to stagnate. On the other hand, the exceptions (e.g., the Epidemic/Venice scenario) suggests that the search space of this problem is characterized, even for a single attacker, by a fitness landscape that allows the EA to refine the search for several generations without stagnating, thus requiring a longer runtime: however, as the number of attackers increases, the landscape probably shows more plateaus where the EA is more likely to stagnate earlier, hence the shorter runtimes. The number of generations required is fairly consistent and small. The computation load is not in the algorithmic engine behind the metaheuristic, but in the time needed to evaluate the behaviour of the protocol via repeated simulations, which is the reason why MaxProp (with its more sophisticated buffer and path optimization) is computationally expensive to test.

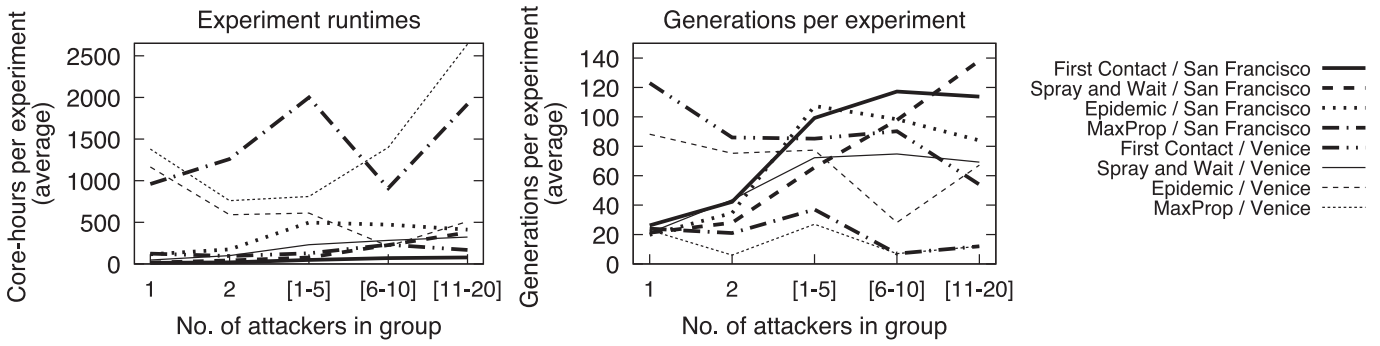

Fig. 7. Runtimes of the evolutionary algorithm until convergence (§ = 50 generations without improvement) for all the experiments: average number of core-hours (left); average number of generations (right).

7. Conclusions

We presented a comparison of alternative search algorithms for computing strong colluding attacks for Delay-Tolerant Networks, and thus uncover the particular weaknesses of DTN protocols: an improved greedy heuristic and a stochastic search algorithm that applies evolutionary principles to generate populations of attack groups in order to minimize the data delivery ratio of the network. One of the main results is that some replicative protocols, generally deemed resilient against possible malicious nodes (due to the implicit redundancy of the messages in the network) can be compromised by a well-crafted attack carried out by a small number of nodes; the most resilient protocol uses end-to-end acknowledgements and contextual shortest-path estimations to optimize message forwarding. These results were obtained on two large-scale urban scenarios characterized by different morphological features. The large experimental campaign presented here shows then that: (a) the two methods are robust enough to be applied to different scenarios, and (b) the conclusions we draw depend only partially on the specific urban map, while some general trends are specific of each routing protocol.

Overall, this work attempts to uncover critical issues of the DTN routing for which, so far, limited, small-scale knowledge was available in the literature. Importantly, our experimental observations shed some light on possible attack scenarios from which DTN routing protocols should be protected at runtime, and can inform further research on the design of proper watchdogs and countermeasures for these attacks, such as mechanisms to detect and cope with black-holes and flooders at the level of new DTN routing protocols.

References

Alcaraz, C., & Lopez, J. (2010). A security analysis for wireless sensor mesh networks in highly critical systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 40*(4), 419–428. doi:10.1109/TSMCC.2010.2045373.

Auerbach, J. E., Iacca, G., & Floreano, D. (2016). Gaining insight into quality diversity. In *Proceedings of the 2016 on genetic and evolutionary computation conference companion* (pp. 1061–1064). ACM.

Brooks, R., Pillai, B., Racunas, S., & Rai, S. (2007). Mobile network analysis using probabilistic connectivity matrices. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 37*(4), 694–702. doi:10.1109/TSMCC.2007.897484.

Bucur, D., Iacca, G., Gaudesi, M., Squillero, G., & Tonda, A. (2016). Optimizing groups of colluding strong attackers in mobile urban communication networks with evolutionary algorithms. *Applied Soft Computing, 40*, 416–426. doi:10.1016/j.asoc.2015.11.024.

Bucur, D., Iacca, G., Squillero, G., & Tonda, A. (2014). The impact of topology on energy consumption for collection tree protocols: An experimental assessment through evolutionary computation. *Applied Soft Computing, 16*, 210–222.

Bucur, D., Iacca, G., Squillero, G., & Tonda, A. (2015). Black holes and revelations: Using evolutionary algorithms to uncover vulnerabilities in disruption-tolerant networks. In A. M. Mora, & G. Squillero (Eds.), *Applications of evolutionary computation*. In *Lecture Notes in Computer Science: 9028* (pp. 29–41). Springer International Publishing. doi:10.1007/978-3-319-16549-3_3.

Burgess, J., Bissias, G. D., Corner, M. D., & Levine, B. N. (2007). Surviving attacks on disruption-tolerant networks without authentication. In *Proceedings of the 8th ACM international symposium on mobile ad hoc networking and computing*. In *MobiHoc '07* (pp. 61–70). New York, NY, USA: ACM. doi:10.1145/1288107.1288116.

Burgess, J., Gallagher, B., Jensen, D., & Levine, B. (2006). MaxProp: Routing for vehicle-based disruption-tolerant networks. In *Infocom 2006. 25th IEEE international conference on computer communications* (pp. 1–11). doi:10.1109/INFOCOM.2006.228.

Choo, F. C., Chan, M. C., & Chang, E.-C. (2010). Robustness of DTN against routing attacks. In *Communication systems and networks (comsnets), 2010 second international conference on* (pp. 1–10).

Fall, K., & Farrell, S. (2008). DTN: An architectural retrospective. *Selected Areas in Communications, IEEE Journal on, 26*(5), 828–836.

Hansen, N., Arnold, D., & Auger, A. (2015). Evolution strategies. In J. Kacprzyk, & W. Pedrycz (Eds.), *Springer handbook of computational intelligence* (pp. 871–898). Springer Berlin Heidelberg.

Islam, K., Shen, W., & Wang, X. (2012). Wireless sensor network reliability and security in factory automation: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 42*(6), 1243–1256.

Jain, S., Fall, K., & Patra, R. (2004). Routing in a Delay Tolerant Network. In *Proceedings of the 2004 conference on applications, technologies, architectures, and protocols for computer communications*. In *SIGCOMM '04* (pp. 145–158). New York, NY, USA: ACM. doi:10.1145/1015467.1015484.

Kempe, D., Kleinberg, J., & Éva Tardos (2015). Maximizing the spread of influence through a social network. *Theory of Computing, 11*(4), 105–147. doi:10.4086/toc.2015.v011a004.

Keränen, A., Ott, J., & Kärkkäinen, T. (2009). The ONE Simulator for DTN Protocol Evaluation. In *Simutools '09: Proceedings of the 2nd international conference on simulation tools and techniques*. New York, NY, USA: ICST.

Lehman, J., & Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation, 19*(2), 189–223.

Li, F., Yang, Y., & Wu, J. (2010). Attack and flee: Game-Theory-Based analysis on interactions among nodes in MANETs. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 40*(3), 612–622. doi:10.1109/TSMCB.2009.2035929.

Mouret, J., & Clune, J. (2015). Illuminating search spaces by mapping elites. CoRR, abs/1504.04909.

Pugh, J. K., Soros, L. B., Szerlip, P. A., & Stanley, K. O. (2015). Confronting the challenge of quality diversity. In *Proceedings of the 2015 annual conference on genetic and evolutionary computation*. In *GECCO '15* (pp. 967–974). New York, NY, USA: ACM.

Rogers, A., David, E., & Jennings, N. (2005). Self-organized routing for wireless microsensor networks. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 35*(3), 349–359. doi:10.1109/TSMCA.2005.846382.

Sanchez, E., Schillaci, M., & Squillero, G. (2011). *Evolutionary optimization: The μGP toolkit* (1st). Springer Publishing Company.

Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2005). Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACMSIGCOMM workshop on delay-tolerant networking* (pp. 252–259). ACM.

Squillero, G., & Tonda, A. (2016). Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences, 329*, 782–799. Special issue on Discovery Science.

Vahdat, A., Becker, D., et al. (2000). Epidemic routing for partially connected ad hoc networks. *Technical Report*. Technical Report CS-200006, Duke University.

Wei, K., Liang, X., & Xu, K. (2014). A survey of social-aware routing protocols in delay tolerant networks: Applications, taxonomy and design-related issues. *IEEE Communications Surveys Tutorials, 16*(1), 556–578. doi:10.1109/SURV.2013.042313.00103.

Zhuang, H., Sun, Y., Tang, J., Zhang, J., & Sun, X. (2013). Influence maximization in dynamic social networks. *2013 IEEE 13th international conference on data mining (ICDM)*.