Ad Hoc Networks 25 (2015) 303–313

Contents lists available at ScienceDirect

Ad Hoc Networks

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

Routing-aware fair contact plan design for predictable delay tolerant networks

J. Fraire *, J.M. Finochietto

Digital Communications Research Lab, Universidad Nacional de Córdoba, CONICET, Argentina

ARTICLE INFO

Article history:
Received 19 February 2014
Received in revised form 15 July 2014
Accepted 23 July 2014
Available online 1 August 2014

Keywords:
Delay tolerant networks
DTN
Contact plan design
Satellite constellations

ABSTRACT

Delay tolerant networks (DTNs) have become a promising solution for extending Internet boundaries to challenged environments such as satellite constellations. In this context, strategies to exploit scarce communication opportunities, while still considering device and application constraints, are still to be investigated to enable the actual deployment of these networks. In particular, the Contact Graph Routing (CGR) scheme has been proposed as it takes advantage of the contact plan, which comprises all future contacts among nodes. However, resource constraints can forbid the totality of these contacts to belong to the contact plan; thus, only those which together meet an overall goal shall be selected. In this article, we consider the problem of designing a contact plan that can provide fairness in link assignment and minimal all-to-all route delay; therefore, achieving equal contact opportunities while favoring end-to-end traffic latency. We formalize this by means of a multi-objective optimization model that can be computationally intractable for large topologies; thus, heuristic algorithms are proposed to compute the contact plan in practice. Finally, we analyze general results from these routines and discuss how they can used to provision valuable contact plans for real networks.

1. Introduction

Delay-Tolerant Networks (DTNs) have received much attention during the last years as they have been proposed for several environments where communications can be challenged by either latency, bandwidth, errors, or stability issues [1]. Even if originally studied to develop a network architecture for the Interplanetary Internet (IPN) [2], DTNs have been recently recognized as an alternative solution for building future satellite applications [3]; in particular, to cope with typical intermittent channels of LEO (Low Earth Orbit) constellation systems [4].

Among the challenges to implement practical DTNs, the definition of a new communication protocol, which does not assume a persistent connectivity between the communication end points, has been addressed by the specification of the Bundle protocol [5], resulting in the availability of several software implementations of the protocol [6,7]. Indeed, traditional protocols like TCP cannot be used for end-to-end communication due to their conversational nature. Neither can traditional routing protocols be employed on environments with intermittent connections, where the network topology is time-varying. To this end, new routing mechanisms have been recently investigated [8–11]. In particular, if topology changes are predictable as in LEO environments [12], the Contact Graph Routing (CGR) [13] scheme is appealing as it takes advantage of the a priori knowledge of the *contact plan* between DTN nodes.

In general, a *contact* can be defined as the opportunity to establish a temporal communication link among two DTN nodes. However, it is possible that a given node may

* Corresponding author. Tel./fax: +54 351 4334147.
E-mail addresses: juanfraire@gmail.com (J. Fraire), jfinochietto@efn.uncor.edu (J.M. Finochietto).

http://dx.doi.org/10.1016/j.adhoc.2014.07.006
1570-8705/

have more than one contact opportunity at a given time but limited or conflicting resources to only make use of one of these opportunities. Furthermore, operational constraints such as power budgets, agency policies, among others might arise. As a result, the contact plan can be thought as a subset of the *contact topology* which comprises all contact opportunities a network of DTN nodes has over a given time window. The design of contact plans has still received little attention, as it is typically assumed that all potential contacts between DTN nodes can belong to the contact plan; in other words, that the contact plan equals the contact topology. Early works [14,15] have focused on contact plans that can enhance the network connectivity for a given topology state, without taking into account the time-evolving nature of the contact topology which enables the transport of data through multiple states. With the advent of DTNs, the design of contact plans that consider time-evolving topologies became much relevant as traffic data can traverse different topology states. Recent works [16,17] have proposed the design of contact plans that can minimize path costs and maximize reliability; however, to the best of our knowledge nor fairness issues neither end-to-end latency have been addressed in the design of contact plans.

In this work, we significantly extend our recent work [18] in order to investigate the design of routing-aware fair contact plans, where all DTN nodes shall have the best possible access to available links among them while optimizing all-to-all route delays. Fairness can be achieved over time by considering the time-evolving nature of the network topology as well as previous link assignments, while route delays can be optimized in a subsequent heuristic stage. Our goal is to compute proper contact plans that can provide equal opportunities to all DTN nodes for the purpose of exchanging data traffic in both direct and routed neighbors. To this end, we formalize this problem, and then propose and evaluate computational efficient algorithms that can be used to design these contact plans.

The paper is structured as follows. Section 2 describes the contact topology and the contact plan models on which fairness and routing criteria metrics are based. We formalize the problem by means of multi-objective optimization problem in Section 3, for which novel algorithmic alternatives are proposed in Section 4. Section 5 evaluates the performance of these strategies in terms of fairness and routing metrics in a general and particular case study respectively. Finally, Section 6 concludes the work and discusses future directions.

2. System model

Nodes from a satellite constellation exhibit a concrete physical trajectory (i.e., orbits), where communication links become both sporadic and foreseeable. In other words, since their position and attitude can be accurately forecasted by precise analytic or numerical models, the same remains true for communication contacts between them. As a result, such constellations can be considered predictable DTNs, where traffic flows in a store, carry, and forward fashion before reaching its destination.

2.1. Contact topology

The aforementioned contact forecast is rendered in the *contact topology*. In order to illustrate the latter, we consider the case of a satellite network with 4 nodes as shown in Fig. 1(a). Here, all nodes are equipped with transponders and antennas enabling cross-links near the pole, while only contacts between nodes 1 and 2, and nodes 3 and 4, are feasible on the equator area. The time-evolving nature of these links can be captured by means of graphs [19], capable of symbolizing links availability over time. This representation can be thought as a finite state machine (FSM) in which each state is characterized by a graph whose arcs, in turn, represent a feasible communication (i.e., contact) between nodes during a period of time. Each state can be identified by $k = 1, 2, \ldots K$ conforming K graphs that comprises the same set of nodes but different arcs among them. Particularly, in the suggested scenario, 3 states can describe the contact topology, representing the communication link evolution during half an orbit time frame.

In particular, a contact topology consists of $p_{k,i,j}$ links between node i and j at state k, where $p_{k,i,j}$ may adopt an integer identifier related to the communication interface. If no contact is feasible, then $p_{k,i,j} = 0$, and $p_{k,i,j} = 1$ if the contact among i and j is possible. For instance, we could state that $p_{1,1,2} = p_{1,2,1} = 1$ and $p_{2,2,3} = p_{2,3,2} = 1$, implying the feasibility of contacts via the pole and equator communication interfaces. Besides, at state $k = 1,\ p_{1,2,3} = p_{1,3,2} = 0$ since no physical link exist between nodes 2 and 3. In general, the contact topology can be defined by a three dimensional *physical* adjacency matrix $[P]_{k,i,j}$.

Finally, the contact topology can be configured to satisfy strict system limitations such as forbidden transmission over a specific geographical area, or specific time interval where interference to geostationary can be provoked, or even operational issues like specific agency policies among others. In the other hand, resource-dependent constraints with impact in the maximum number of interfaces are the focus of research of this article and defined in Section 2.2.

2.2. Contact plan

The contact topology assumes no resource limitation on nodes as it is based only on the feasibility of physical communication. However, nodes may only be able to make use of limited contacts at a given time, even if more are potentially available. This can be due to power budget limitations, hardware designs, interference requirements, among others. As a result, the maximum number of simultaneously active interfaces can be thought as a restriction to the contact topology $[P]_{k,i,j}$. The problem then lies in selecting the most appropriate set of possible contacts in a time-evolving network with a certain criteria subject to restrictions or limitations. Consequently, we define an integer *contact* matrix $[C]_i$ constituted by the maximum c_i contacts a node i can simultaneously maintain. For example, if we

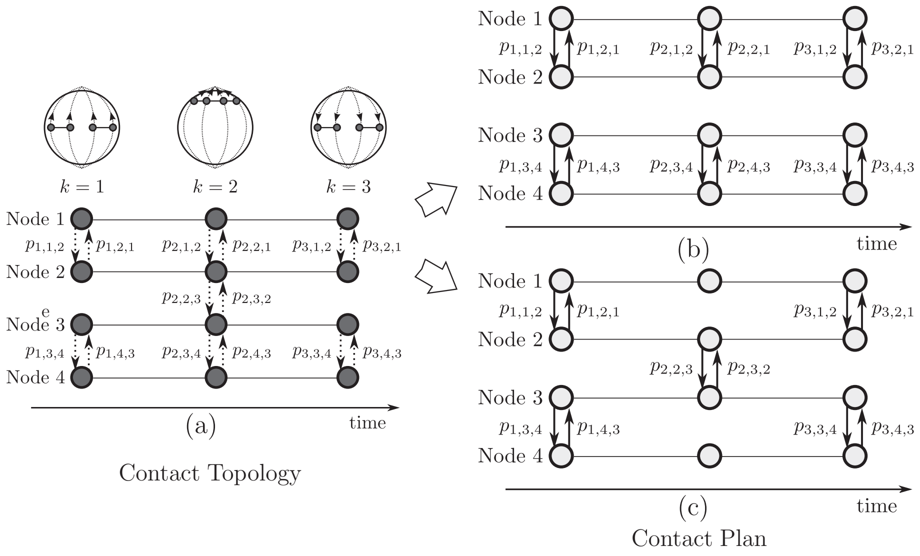

Fig. 1. Representation of evolving topologies: contact topology and contact plan.

assume $c_i = 1 \forall i$, and particularly for $i = 2$ in the case of Fig. 1(a), a decision must be taken for nodes 2 and 3 at $k = 2$. Indeed, two possible solutions, as illustrated in Fig. 1(b) and (c), are valid enabling either $p_{2,1,2}$, $p_{2,2,1}$, $p_{2,3,4}$, and $p_{2,4,3}$ in (b) or $p_{2,2,3}$ and $p_{2,3,2}$ in (c). If the first topology is chosen, this results in the maximum number of contacts, while if the second one is selected a more fair and connected network is obtained given that a contact between 2 and 3 is scheduled. Both solutions are defined as feasible *contact plans* the network can implement with the specified resources. In particular, we investigate the single contact per node constraint scenario ($c_i = 1 \forall i$).

The search for a general answer to the latter is not trivial, specially as nodes, contacts, and states increase, being this the principal motivation of the present work. Our first work [18] focused only on the exploration of an appropriate link selection method aiming to maximize the fairness in the resulting contact plan, while this work extends those ideas to include routing knowledge to improve the contact plan overall end-to-end latency. This plan is symbolized by $[L]$ whose $l_{k,i,j}$ elements represents the chosen unidirectional arcs from node i to j at state k. Therefore, $[L] \subseteq [P]$. Strictly speaking, $[P]$ stands for the potential contact set, while $[L]$ for the selection over $[P]$. Then, $[P]$ proves to be one of the inputs while $[L]$, the result of the desired design procedure.

For sake of simplicity, we assume communication links whose bandwidth is taken constant and equal – capacity is then analogous to contact time. Furthermore, we design the contact plan to serve an uniformly distributed traffic pattern in the network, where all nodes exchange traffic which each other. Nevertheless, the proposed models and algorithms can easily be generalized to fit different traffic patterns.

2.3. Routing awareness

Increasing the fairness in contact distribution among the nodes can provide a fair distribution of an all-to-all traffic pattern. However, real routing knowledge can be taken into account to improve the contact plan design. The use of route information can enrich the link selection criteria by considering parameters (i.e. route delay) that better assess the design goal: to service network traffic. In other words, a fair contact distribution approach focuses only on the link assignment problem, while considering routes allows us to refine such a decision favoring traffic latency.

In particular, we adopt the *space–time* routing framework proposed by [19] for predictable *store, carry and forward* networks. The algorithm devised is closely related to the well-known Floyd–Warshall algorithm that computes the shortest delay path between all pair of vertices in a given graph [20]. This algorithm output is analog to that of the CGR [13] routing scheme for predictable DTNs – disregarding network congestion, as both are equivalent to Dijkstra's shortest path method. The output of the routing algorithm is a set of K matrices $[R]_{k,i,j}$ that specifies both the next hop in the path to the final destination and the delay $d_{k,i,j}$ for each destination node in each k state of the contact plan. In Fig. 2 both $[R]$ matrices can be seen for the two possible contact plans. It is clear from the routing tables that contact plan (a) exhibits a limited route quantity, being this evidence of broken topologies valuable for the design criteria. Also, in general, towards the end of the contact plan time frame, the routes tends to become more scarce.

2.4. Metrics

In order to quantify the properties of contact plans, we propose the fairness and routing metrics shown in Fig. 3.

In terms of fairness, the *Raj Jain's fairness index* (JI) determines the total time a given contact (out of the $(i * j)$ total) is enabled $(l_{k,i,j} * t_k)$ throughout the contact plan. The result ranges from $1/(i * j)$ for the worst allocation case to 1 for a maximum fairness. This index has the property of measuring all allocations in contrast with the

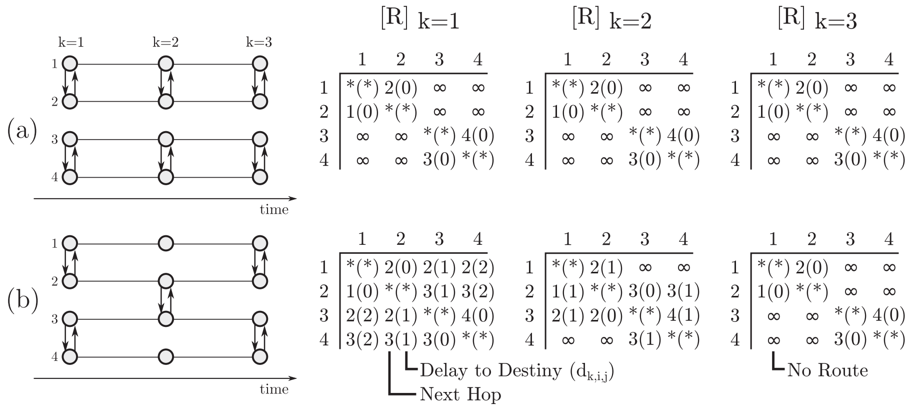

Fig. 2. Space–time routing for two possible contact plan.

traditional min–max fairness metric. For the contact plans in Fig. 2(a) and (b) the indexes are 0.25 and 0.5 respectively implying that the second topology is more fair as expected.

Regarding routing metrics, *Max Average Delay* (MD) is the maximum average route delay among all k, between all node pairs in the contact plan, accounting only for non-infinite $d_{k,i,j}$. Routes with infinity value are considered nonexistent, where the accumulated time that this remains true for all node pairs along the topology is measured by the *unrouted time* (UT) metric. It is interesting to note the relation between MD and UT where in the case UT increases (i.e. fewer routes exists), MD can improve as it evaluates less source–destination pairs. For instance, for the topologies in Fig. 2, the contact plan (a) proves to have an MD of 0 (since all delay $d_{k,i,j} = 0$) while (b) has an MD of 2 (since $d_{1,1,4} = d_{1,4,1} = 2$). It might seem that (a) is better, but a respective UT metric of 24 and 12 for (a) and (b) evidences the lack of valid routes in the first topology. In other words, the trade-off is having fewer but good routes or many with higher delays.

In general, we can state that a good contact plan design delivers the highest JI, and the lowest UT and MD. However these metrics results incommensurable making the decision difficult when conflicts arise between them. We further discuss this topic in Section 3.

3. Problem formulation

As discussed early, the main goal is to obtain contact plan with a high fairness index (JI) that guarantees the best all-to-all link distribution while also providing existence (UT) of low route delays (MD) to assure good service to the assumed all-to-all traffic pattern. However, the nature of these metrics result incommensurable and even conflicting when requiring simultaneous optimization. This problem can be then formalized as multi-objective optimization one, whose general solution was analyzed by Pareto [21], concluding the existence of a set of possible solutions (Pareto-optimal) that are optimal in a wider sense; where no other solutions in the search space are superior to each other when considering all objectives.

Our problem is to find a contact plan represented by a solution vector L of the form:

$$L = [l_{1,1,1}, l_{1,1,2}, \ldots l_{k,i,j}]^T \quad \forall k,i,j : \quad l_{k,i,j} \in \{0,1\} \tag{1}$$

that satisfies the resource restrictions related to the fact that nodes can only establish one contact at each state k, as described in detail in [18], and optimize the objective function given by:

$$F_{obj} = [min : MD, min : UT, max : JI]^T \tag{2}$$

Solving the problem in this way quickly becomes computationally intractable as the addition of nodes, contacts, or states exponentially increases the required resolution time. For instance, a simple topology of $k = 10$ states, with 10 nodes, and a probability of link existence in the contact topology of 0.3, allows for proximately 300 feasible contacts that can be chosen to be enabled or not, deriving in a solution space of 2^{300} possible topologies to evaluate. Despite we implemented a exhaustive enumeration – brute force – algorithm of this kind for comparison and evaluation purposes in Section 5, it lacks of usage in real topology planning. In the next section, we propose an algorithmic approach that provides sub-optimal solutions using heuristics methods.

$$JI = \frac{(\sum_i \sum_j \sum_k l_{k,i,j} * t_k)^2}{(i*j) * \sum_i \sum_j (\sum_k l_{k,i,j} * t_k)^2} \qquad MD = max_{i,j}\left(\frac{\sum_k d_{k,i,j}}{q_d}\right) \quad \forall d_{k,i,j} < \infty \qquad UT = \sum_i \sum_j \sum_k t_k \quad \forall k,i,j : d_{k,i,j} = \infty$$

Fig. 3. Contact plan metrics.

4. Contact plan design algorithms

In order to tackle the design of both fair and routing aware contact plans for delay tolerant networks we propose a two stage procedure that in the first place determines a fair contact distribution among nodes, and then, improves link assignment to benefit overall routing metrics. This can be achieved by means of the Fair Contact Plan (FCP) [18] algorithm in the first place, and iteratively refining the resulting contact plan as the route algorithm is known in advance. As part of this second stage, the JI fairness metric is thus combined with MD and UT routing metrics to obtain a balanced solution via different algorithms. Despite the proposed algorithms can be easily adapted to fit any routing method, we choose adopt the Floyd–Warshall based routing scheme described in Section 2, which provides an all-to-all lowest delay routes for each topological state.

Fig. 4 depicts this two stage design flow, where after obtaining a first contact plan L' through the FCP algorithm, heuristics such as Steepest Descent (SD), First Improvement (FI) and Simulated Annealing (SA) are used to explore solutions that can provide better routing metrics, and generate the final contact plan L.

4.1. Fair contact plan algorithm

The contact plan shall limit each node to handle one single contact at each state ($[C]_i = 1\ \forall i$); therefore, the design of the contact plan can also be considered a matching problem, which has to be solved at every state. In our context, a matching consists of a set of contacts such as no node is assigned more than one contact in a given state. Since each state can be associated to a general graph, we need to consider the general matching problem which can be solved by means of the Blossom algorithm [22,23].

In Blossom, the matching is constructed by iteratively improving an initial empty matching along augmenting paths in the graph. At each iteration the algorithm either finds an augmenting path, finds a blossom and recurses onto the corresponding contracted graph, or concludes there are no augmenting paths. If there were no cycles, the algorithm reduces to standard bipartite matching. However, the Blossom algorithm only searches for perfect matchings, implying all vertex must be covered and requiring an even number of nodes. Particularly, our problem pursues the maximum weight in the more general case of – not necessarily perfect – maximum matching (i.e. we are willing to leave an uncovered vertex as long as the sum of weights is maximized). In [24] a reduction of the maximum (non-perfect) matching to a perfect matching problem is proposed by using a $G^*(V^*, E^*, w^*)$ auxiliary network of $G(V, E, w)$ with equivalent vertex $v^* = v$, edges $e^* = e$, and weights $w^* = w$ in order to consider a summation of $G + G^* = G'$ where $V' = V \cup V^*$, $E' = E \cup E^* \cup \{vv^*$: $v$is in V and v^* is in $V^*\}$, and $w' = \{w$ when in E, w^* when in E^*, and 0 when $vv^*\}$. Solving the summation by a perfect matching algorithm as Blossom delivers a non-perfect weighted matching of G inside G' as shown in Fig. 5. This reduction doubles the size of the graph and includes as many auxiliary arcs as nodes in the main graph. The Blossom solution of the reduced graph satisfies the proposed model becoming both, the weighted matching and the reduction, part of the suggested FCP procedure under the *ReducedBlossom*() routine in Algorithm 1.

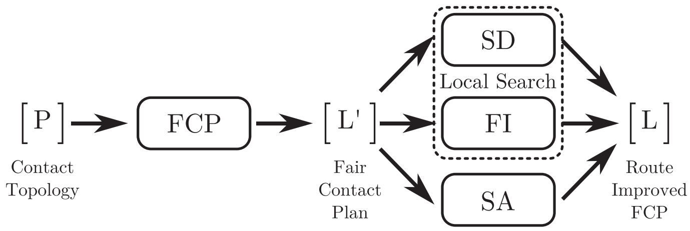

Fig. 4. Contact design flow.

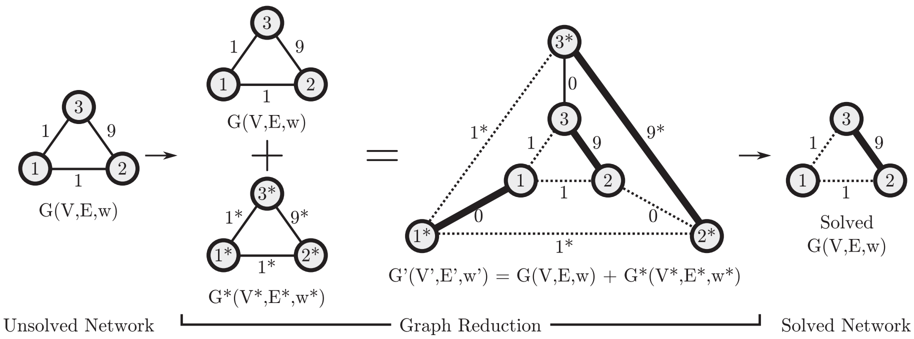

Fig. 5. Reduction of the maximum matching to a perfect matching.

Algorithm 1. Fair Contact Plan (FCP) Algorithm

input : Contact Topology $[P]$ of size $K \times N \times N$ and State Time $[T]_k$
output: Contact Plan $[L]$ of size $K \times N \times N$
$DCT_{i,j} \longleftarrow 0 \quad \forall i,j;$
for $k \leftarrow 0$ *to* K **do**
  $[W]_{k,i,j} \longleftarrow DCT_{i,j} \quad \forall i,j$ `ReducedBlossom`$([P]_k,[L]_k,[W]_k)$;
  if $[L]_{k,i,j} = 0$ **then**
    $DCT_{i,j} \longleftarrow DCT_{i,j} + t_k \quad \forall i,j$

Since our goal is to provide fairness over time, contacts (i.e., arcs) can be weighted as a function of the time a link between a pair of nodes has not been assigned either due to no physical feasibility in $[P]$ or previous FCP decisions. As a result, FCP aims at finding $[L]$ with a sub-set of maximum fairness weighted arcs restricted to one contact per node for a given state k. By means of a dynamic programming approach, arc weights $[W]$ are internally calculated on a per state bases as the accumulated disabled contact time ($DCT_{i,j} = t_{k1} + t_{k2} + \ldots + t_{kn}$); namely, the time that a contact between i and j has not been enabled in the past states. In other words, the more time a link is not in the resulting contact plan $[L]$, the more weight it gets, and the more chance it has to be chosen in future states. It is worth noting that the Blossom algorithm provides optimal (i.e. fair) solutions on a per state basis, while FCP combines these solutions to generate one that considers fairness over all states. Therefore, the FCP solution can be suboptimal and potentially improved by other algorithms as it will discussed in the following subsections. Algorithm 1 details the complete procedure, and Fig. 6 illustrates FCP behavior with the example contact topology of Section 2.

Regarding final algorithm complexity, several efficient implementations such as [25] solves Blossom algorithm in $O(n^2 l)$, where n is the node number and l is the arc quantity. Given the proposed reduction to the graph, and the need to perform this evaluation on all k states, it can be demonstrated that the final complexity of FCP becomes $O(2kn^2(2l + n))$.

4.2. Local search algorithms

The contact plan delivered by FCP is characterized by equitably distributed contact both among node pairs and in time. The latter results beneficial for all to all traffic patterns routed by general shortest delay algorithms. Nevertheless, if the specific routing algorithm is well known, the resulting contact plan $[L]$ can be further refined to keep fairness metric within a controllable range while improving routing metrics. To this end, we first propose the use of two popular local search algorithms: First Improvement (FI) and Steepest Descent (SD) to address this goal.

The FI algorithm, detailed in Algorithm 2, starts from the contact plan delivered by FCP to continue with a local search for other plans with better route delay metrics, evaluated by means of the *FloydWarshall* routine. The latter provides a routing matrix $[R]_{k,i,j}$ from which these metrics can be derived and compared. The first neighbor that is found to improve the maximum average delay (MD) and unrouted time (UT) route metrics (*MaxD* and *UnRT* implementations), while not degrading the Jain Index (JI), is adopted as the refined plan. This is analogous to find a dominant solution vector as per Pareto optimality criteria [26]. These routines executes for a maximum number of iterations.

In the other hand, the SD algorithm, described in Algorithm 3, conforms a neighborhood ($[L]'_n$) of a fixed size *MaxNeighbors* for each iteration. Among this, the best plan is chosen as the best neighbor given that the MD and UT improves and the JI metrics does not degrade. If no best neighbor can be found (*NoBestNeighbor* flag), the SD routine exits. SD allows to better explore nearby possibilities that FI might had disregarded as adopting the first improvement approach, but can exit prematurely if the explored neighborhood shows no better contact plan. Fig. 7(a) and (b) illustration compares FI and SD approach graphically with a single symbolic metric scenario.

Algorithm 2. First Improvement (FI) Algorithm

input : Contact Topology $[P]$ of size $K \times N \times N$ and State Time $[T]_k$
output: Contact Plan $[L]$ of size $K \times N \times N$
$[L] \longleftarrow$ `FCP`$([P],[T])$;
$[R] \longleftarrow$ `FloydWarshall`$([L])$;
$[BMD, BUT, BJI] \longleftarrow$ [`MaxD`$([R])$, `UnRT`$([R])$, `JainI`$([L])$];
for $i \leftarrow 0$ *to* $MaxIterations$ **do**
  $[L]' \longleftarrow$ `GetNeighbor`$([L])$;
  $[R]' \longleftarrow$ `FloydWarshall`$([L]')$;
  $[CMD, CUT, CJI] \longleftarrow$ [`MaxD`$([R]')$, `UnRT`$([R]')$, `JainI`$([L]')$];
  if $(CMD <= BMD)\&(CUT <= BUT)\&(CJI >= BJI)$ **then**
    $[L] \longleftarrow [L]'$;
    $[BMD, BUT, BJI] \longleftarrow [CMD, CUT, CJI]$;

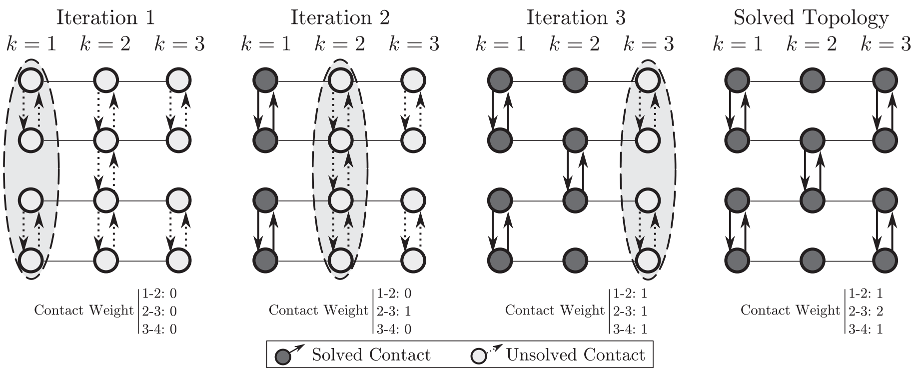

Fig. 6. Fair contact plan algorithm behavior.

In both algorithms the MD, UT, and JI metrics are computed for each state storing the *current* values in *current max-average delay* (CMD), *current unrouted time* (CUT), and *current Jain-index* (CJI), and the *best* ones in *best max-average delay* (BMD), *best unrouted time* (BUT), and *best Jain-index* (BJI). The best contact plan is always kept in $[L]_{k,i,j}$ until the return of the routine.

Algorithm 3. Steepest Descent (SD) Algorithm

```
input : Contact Topology [P] of size K × N × N and State Time [T]_k
output: Contact Plan [L] of size K × N × N
[L] ⟵ FCP([P],[T]);
[R] ⟵ FloydWarshall([L]);
[BMD, BUT, BJI] ⟵ [MaxD([R]), UnRT([R]), JainI([L])];
for i ← 0 to MaxIterations do
    for n ← 0 to MaxNeighbors do
        [L]'_n ⟵ GetNeighbor([L]);

    for n ← 0 to MaxNeighbors do
        [R]'_n ⟵ FloydWarshall([L]'_n);
        [CMD, CUT, CJI] ⟵ [MaxD([R]'_n), UnRT([R]'_n), JainI([L]'_n)];
        if (CAD <= BAD)&(CUT <= BUT)&(CJI >= BJI) then
            [L] ⟵ [L]'_n;
            [BMD, BUT, BJI] ⟵ [CMD, CUT, CJI];
    if NoBestNeighbor then
        exit
```

Both FI and SD require neighbors topologies in each iteration that satisfies the single contact constraint in order to meet the system model requirement (i.e $[C]_i = 1\ \forall i$). Despite that in both cases the initial contact plan delivered by FCP satisfies this by definition, this must remain true for all further neighbors generation. This is a achieved by a branch-and-exchange methodology that still uses Blossom algorithm as driver. In particular, we detect nodes with two or more links in the contact topology $[P]_{k,i,j}$ for a given state since this implies that only one of them has been chosen to finally conform the FCP contact plan $[L]_{k,i,j}$. Once detected, Blossom is forced – in this particular state – to adopt the unchosen contact. To this end we knowingly and temporarily disable the chosen link $p_{k,i,j}$ in the contact topology and executes the Blossom algorithm only for this state, deriving in a yet fair new state, satisfying the single interface restriction, but with a different link selection. We implement this procedure in the *GetNeighbor* routine providing a valid $[L]'_{k,i,j}$ neighbor in the $[C]_i = 1\ \forall i$ solution space.

Despite these algorithms improve initial topologies, they stop when no best neighbor can be found. This only guarantees finding local optimum solutions which drastically reduces the search space, hence limiting the probability of finding the global optimum solution.

4.3. Simulated annealing algorithm

The local search algorithm exposed are fairly simple and effective as far as good possible solutions are found in the proximity of the initial topology delivered by FCP, nevertheless this is not necessarily the general case. When aiming at exploring the complete solution the problem complexity scales significantly due to the large search space involved as stated in Section 3.

Algorithm 4. Simulated Annealing (SA) Algorithm

```
input : Contact Topology [P] of size K × N × N and State Time [T]_k
output: Contact Plan [L] of size K × N × N
[L] ⟵ FCP([P],[T]);
[R] ⟵ FloydWarshall([L]);
[BMD, BUT, BJI] ⟵ [MaxD([R]), UnRT([R]), JainI([L])];
for i ← 0 to MaxIterations do
    [L]' ⟵ GetNeighbor([L]);
    [R]' ⟵ FloydWarshall([L]');
    [CMD, CUT, CJI] ⟵ [MaxD([R]'), UnRT([R]'), JainI([L]')];
    if (CMD <= BMD)&(CUT <= BUT)&(CJI >= BJI) then
        [L]_best ⟵ [L] ⟵ [L]';
        [BMD, BUT, BJI] ⟵ [CMD, CUT, CJI];
    else
        if exp(-ΔMD/T_MD) * exp(-ΔUT/T_UT) * exp(ΔJI/T_JI) > rand(0,1) then
            [L] ⟵ [L]';
            [T]_i ⟵ [T]_i - 1    : i = JI, MD, UT
        else
            [L] ⟵ [L]_best;
[L] ⟵ [L]_best;
```

In this context, Simulated Annealing (SA) [27,28] is a well known meta-heuristic for locating approximations of a global optimum solution problem, specially in discrete solution spaces. In contrast to FI and SD algorithms, SA can accept non-improving solutions as a way of increasing the search space. In particular, the objective of SA is to bring the system from a start point to a minimum state of energy. To this end, the algorithm probabilistically moves from one state to another guided by the energy difference between the states, with the chance of accepting any neighbor in order to avoid getting stuck in local optimums as demonstrated for FI and SD in Section 4.2.

The chance of acceptance is driven by a probability function A of previous and actual state energies e and e' (namely the variation of the objective vector $(\Delta MD, \Delta UT, \Delta JI)$, and the corresponding annealing temperature T_i, this is:

$$A() = exp\left(-\frac{\Delta MD}{T_{MD}}\right) * exp\left(-\frac{\Delta UT}{T_{UT}}\right) * exp\left(\frac{\Delta JI}{T_{JI}}\right) \quad (3)$$

The corresponding temperature T_i decreases as worst neighbors are adopted with the advance of the iterations; therefore, the same happens with the probability A of accepting worst solutions. This allows to perform deep general searches in the solution space specially at the beginning of the iterations turning to more specific and fine grained local search towards the end. The decreasing of T_i is called *annealing schedule* [29] and chosen to be lineal for each individual $[T] = [T_{MD}, T_{UT}, T_{JI}]$ in our particular implementation as suggested by the method of Suppapitnarm and Parks in [30]. Finally, we adopted a return-to-base

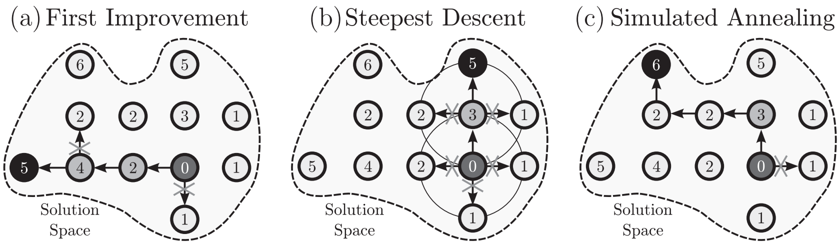

Fig. 7. FI, SD, and SA illustration.

strategy [31] meaning that the rejection of a solution implies a return to the latest best solution known and archived. Fig. 7(c) illustrates the SA algorithm behavior and compares with FI and SD local search detailed in Section 4.2. Further comparisons are provided in Section 5.

Algorithm 4 details the proposed SA algorithm for designing contact plans with routing knowledge. Despite FI, SD and SA algorithms account for the same improvement criteria $((CMD <= BMD)\&(CUT <= BUT)\&(CJI >= BJI))$, they can get to different sub-optimal solutions of (MD, UT, JI). In particular [31], states that SA is capable of reaching the Pareto frontier given the proper iterations are configured.

5. Evaluation

5.1. General analysis

In order to evaluate the proposed algorithms, we consider a set of random contact topologies $[P]_{k,i,j}$, and evaluate the delivered contact plan $[L]_{k,i,j}$ in terms of their metrics. Random topologies are generated by varying the link density (LD), namely, changing the average number of available contacts at each state, which is then stored in $[P]_{k,i,j}$. Indeed, the higher the LD, the more contact population in $[P]$, and the greater the solution search space. LD is adjusted from 8 to 20 with steps of 1, while static parameters throughout the

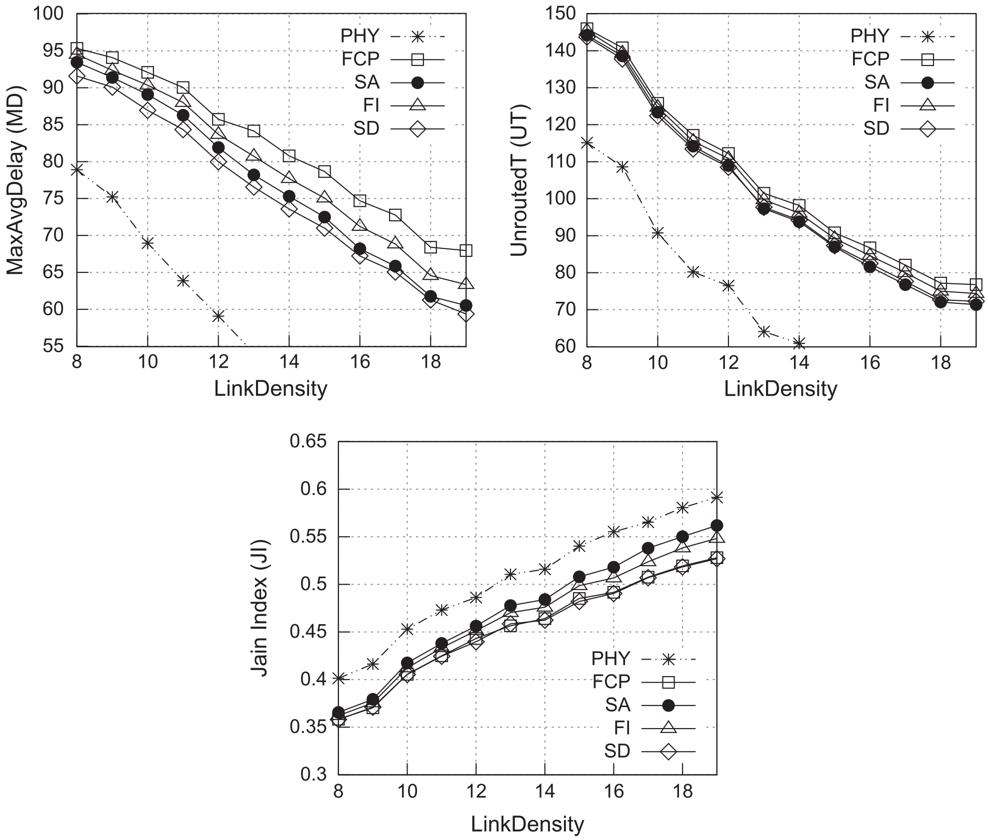

Fig. 8. Evaluation of contact plan design algorithms for different link densities.

topologies are: nodes quantity ($N = 6$), states ($K = 10$), and states times ($10 < t[k] < 20\ \forall k$). The SA algorithm parameters are set to a maximum of 3000 iterations with starting $T_i = 2000\ \forall i$, while FI, to 500 iterations, and SD is configured to continue neighbor search until no best neighbor can be found.

For each LD step, 500 contact topologies $[P]_{k,i,j}$ are generated and solved by the algorithms, whose resulting contact plan $[L]_{k,i,j}$ metrics (MD, JI and UT) are averaged and plotted in Fig. 8. Besides results from these algorithms, the metrics of the contact topologies (PHY) are shown for comparing against the case where $[C]_i$ is not constrained (i.e., nodes can setup more than on contact per state). Finally, the FCP only metrics helps to evaluate how the second stage algorithms (FI, SD and SA) improve the FCP solution.

From these results it is clear that, despite the efficiency of FCP discussed in [18], applying the one contact restriction penalizes all metrics when compared to the contact topology itself, labeled as PHY. This phenomenon is more important as the LD increases since more contacts need to be disabled to satisfy the $C[i] = 1\ \forall i$ restriction. In between PHY and FCP results, stands FI, SD, and SA ones, demonstrating how the algorithms are able to push the initial FCP solution towards a better metric space with better routing performance. The SA algorithm clearly outperforms FI and SD as it is able to improve JI and minimize UT, while only SD is the best at minimizing the MD metric but without enhancing the JI at the same time.

In general, SA offers the best metric improvement combination; nevertheless, FI and SD prove to improve FCP metrics with less computational requirements than SA. However, it is difficult to compare averaged results considering the Pareto frontier in order to better submit the resulting metrics to a formal optimality criteria. In Section 5.2 we propose a particular case study for the latter analysis.

5.2. Case study

Several reasons support the linear flight formation constellation (i.e. train alike) of nodes in LEO satellites. Among them, the fact that spacecrafts perceive quite the same gravity perturbations allowing to save propellant for station-keeping. Besides, it is a desirable formation from a launcher point of view as it does not require orbit transfer to deploy all nodes. Last but no least, from a mission perspective, it allows to provide stereoscopic earth observation images.

These reasons drove the design of the A-Train satellite constellation [32]. Despite the A-Train does not count with inter-satellite links (ISL), it opened the way to consider an autonomous DTN node cluster topology for which we consider the resource limitation detailed in this work. In particular, we propose a 4 node linear network with single contact restriction in order to evaluate how the algorithms can support network planning optimization. The constraint impacts on the two intermediate nodes that have two simultaneous feasible contact available. We also fractionate the topology state in 8 to allow different link assignment opportunities. For simplicity, no ground segment is included in the contact topology to better focus on the

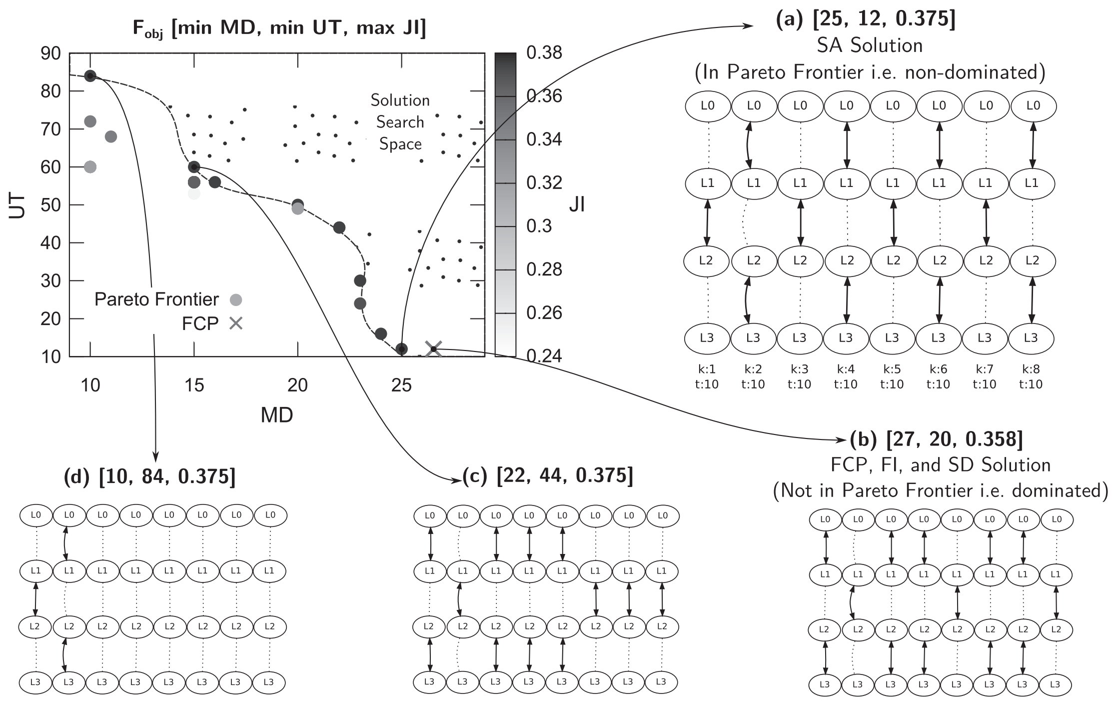

Fig. 9. Evaluation of contact plan design algorithms for linear formation.

train formation allocation. Fig. 9 illustrates the solution search space, the Pareto frontier for this particular problem, and a few solution examples to discuss.

In the plot of Fig. 9 the valid solution space of $F_{obj}[MD, UT, JI]$ was determined via an enumeration process and illustrated with the Pareto frontier marked as dashed line. Despite that intuitively plotting Pareto frontier is a current research topic [33], we decided to plot *MD* and *UT* in *x* and *y* axis respectively, while *JI* value is coded in the gray-scale intensity of the dot. As shown in figure (b), FCP, FI, and SD provide a valid contact plan with objectives values of $[27, 20, 0.358]$, which is a Pareto dominated set (i.e. not in the optimality frontier). This implies that there exists one or more solutions that can dominate this set, and indeed, the $[25, 12, 0.375]$ set provided by SA algorithm in figure (a) does this ($25 < 27$, $12 < 20$, and $0.375 > 0.358$) as it lies in the Pareto frontier.

Other solutions in the frontier are also shown in Fig. 9(c) and (d). Despite these solutions satisfy the Pareto optimality criteria [21], non of them result suitable for a real topology planning. This is mainly because the UT metric goes far away from the initial value of 20, that as explained in Section 2.3, allows for better (lower) MD, while JI however can be as good as 0.375 which is the maximum value possible in the train topology (arcs 0–2, 0–3 and 1–3 can never be physically enabled). Solution (d) is clearly a bad choice; however, it evidences fairly high MD and JI metrics.

The former situations are what Pareto studies define as *decision makers* (DM), which besides algorithms can be used to generate valuable solutions (i.e., contact plans), require humans to finally select the best solution. As a result, we argue that the proposed design flow can be used for this purpose. In particular, we have proven that SA can reach a Pareto frontier while also making the correct topological decision.

6. Conclusion

In this article we introduced the problem of the design of contact plans for predictive delay tolerant networks with constrained resources, such as satellite constellations. In particular, this design was proposed to be driven by fairness in terms of link assignment and by minimal all-to-all routes delays, so as to provide valuable contact plans to route data traffic. We formalized the problem facing the need for computationally effective routines to support real network planning that can be used by humans to make final decisions. To this end, we proposed and evaluated different algorithmic approaches.

Traditional local search algorithms proved to provide reasonable solutions with minimal computing complexity, while simulated annealing meta-heuristic allowed to further explore the solution space; thus, improving the overall contact plan performance. These algorithms were evaluated in general random topologies and in a special scenario of particular interest in low earth orbit constellations for earth observations missions.

This work only considers link assignment fairness and routing delays as the contact plan design criteria, we left as further work the inclusion of traffic predictions to provide even more suitable topologies. The latter is of interest as traffic in satellite constellations can be highly predictive as the system behavior tends to be driven by a mission control center in a centralized fashion.

References

[1] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proceedings of ACM SIGCOM, Karlsruhe, Germany, 2003, pp. 27–34.
[2] S. Burleigh et al., Delay-tolerant networking: an approach to interplanetary internet, IEEE Commun. Mag. 41 (6) (2003) 128–136.
[3] C. Caini et al., Delay and disruption tolerant networking: an alternative solution for future satellite networking applications, Proc. IEEE 99 (2011) 1980–1997.
[4] C. Caini et al., in: Personal Satellite Services, Springer, 2011, pp. 186–198 (Chapter DTN for LEO Satellite Communications).
[5] K. Scott et al., Bundle protocol specification, RFC 5050, Internet Eng. Task Force (IETF) (2007).
[6] DTN2. <http://www.dtnrg.org/wiki/Dtn2Documentation>.
[7] Interplanetary Overlay Network (ION). <http://sourceforge.net/projects/ion-dtn/>.
[8] A. Lindgren et al., Probabilistic routing protocol for intermittently connected networks, RFC 6693, Internet Eng. Task Force (IETF) (2012).
[9] J. Burgess, et al., Maxprop: routing for vehicle-based disruption-tolerant networks, in: IEEE INFOCOM, 2006, pp. 1–11.
[10] A. Balasubramanian, et al., DTN routing as a resource allocation problem, in: ACM SIGCOMM, 2007, pp. 373–384.
[11] T. Spyropoulos, et al., Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: ACM SIGCOMM, 2005, pp. 252–259.
[12] C. Caini, et al., Application of contact graph routing to LEO satellite DTN communications, in: IEEE ICC, 2012, pp. 3301–3305.
[13] S. Burleigh, Contact graph routing, Draft, Internet Eng. Task Force (IETF) (2010).
[14] M. Noakes et al., An adaptive link assignment algorithm for dynamically changing topologies, IEEE Trans. Commun. (1993) 694–706.
[15] H. Chang et al., Topological design and routing for low-earth orbit satellite networks, in: GLOBECOM, IEEE, 1995, pp. 529–535.
[16] M. Huang et al., Cost-efficient topology design problem in time-evolving delay-tolerant networks, in: GLOBECOM, IEEE, 2010, pp. 1–5.
[17] M. Huang et al., Topology design in time-evolving delay-tolerant networks with unreliable links, in: GLOBECOM, IEEE, 2012, pp. 5296–5301.
[18] J. Fraire et al., On the design of fair contact plans for delay tolerant networks, in: International Conference on Wireless for Space and Extreme Environments (WiSEE), IEEE, 2013, pp. 1–7.
[19] S. Merugu, et al., Routing in Space and Time in Networks with Predictable Mobility, Tech. Rep., Georgia Institute of Technology, 2006.
[20] R. Floyd, Algorithm 97: shortest path, CACM., 1962.
[21] V. Pareto, Cours d'économie politique, Rouge: Lausanne 1 and 2.
[22] J. Edmonds, Maximum matching and a polyhedron with 0–1 vertices, J. Res. Nat. Bur. Stand. 69 (1965) 125–130.
[23] J. Edmonds, Path, trees, and flowers, Can. J. Math. 17 (1965) 449–467.
[24] G. Schäfer, Weighted matchings in general graphs, Master's thesis, Universitát des Saarlandes, Saarbrücken, Germany, 2000.
[25] V. Kolmogorov, Blossom V: a new implementation of a minimum cost perfect matching algorithm, MPC 1 (2009) 43–67.
[26] C.L. Hwang et al., Multiple Objective Decision Making, Methods and Applications: A State-of-the-Art Survey, Springer-Verlag, 1979.
[27] S. Kirkpatrick et al., Optimization by simulated annealing, Science 220 (13) (1983) 671–680.
[28] V. Černý, Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm, JOTA 45 (1985) 41–51.
[29] W. Press et al., Numerical Recipes: The Art of Scientific Computing, third ed., vol. Section 10.12, Cambridge University Press, 2007.
[30] A. Suppapitnarm et al., Simulated annealing: an alternative approach to true multi-objective optimization, Eng. Opt. 33 (2000) 59–85.
[31] B. Suman et al., A survey of simulated annealing as a tool for single and multiobjective optimization, JORS 57 (2006) 1143–1160.

[32] NASA: Introducing A-Train, April 2012 (10.26.10). <http://atrain.nasa.gov/>.
[33] G. Agrawal, et al., Intuitive Visualization of Pareto Frontier for Multi-Objective Optimization in n-Dimensional Performance Space, Tech. Rep., Buffalo University, 2004.

Juan A. Fraire received the Telecommunications Engineering degree at the Instituto Universitario Aeronáutico and is completing his PhD on Engineering Sciences in the National University of Córdoba (UNC). His research focuses on predictable DTN networking topology design for space-oriented applications. He is the communications protocol architect for Servicios Tecnológicos Integrados Company where he is developing DTN solutions for earth-observation LEO constellation projects defined in the Argentinian Space Agency (CONAE) National Space Plan.

Jorge M. Finochietto holds a MS and PhD in Electonics Engineer from Universidad Nacional de Mar del Plata, Argentina, and from Politecnico di Torino, Turin, Italy, respectively. His research interests are in the field of performance evaluation, high-speed networking and switching, and optical and wireless networks. He is currently Associate Professor at Universidad Nacional de Córdoba, Argentina (UNC), and he is also Adjunct Researcher at the National Research Council (CONICET) of Argentina. From 2005 to 2007, he was a post doc student at the Telecommunication Network Group of Politecnico di Torino. Since 2007 he is member of the Digital Communications Group at UNC. He has been involved in several national and international research projects in the fields of communication networks. He has co-authored over 50 papers published in international journals and presented in leading international conferences.