POWVER

Technical Report 2020-11

Title: **Distributed Contact Plan Design for GNSSs**

Authors: Zhibo Yan, Juan A. Fraire, Kanglian Zhao, Hongcheng Yan, Pablo G. Madoery, Wenfeng Li, Hong Yang

Report Number: 2020-11

ERC Project: Power to the People. Verified.

ERC Project ID: 695614

Funded Under: H2020-EU.1.1. – EXCELLENT SCIENCE

Host Institution: Universität des Saarlandes, Dependable Systems and Software
Saarland Informatics Campus

Published In: IEEE Trans. Aerosp. Electron. Syst. 56 (1)

Distributed Contact Plan Design for GNSSs

Zhibo Yan*, Juan A. Fraire†‡, Kanglian Zhao*, Hongcheng Yan§, Pablo G. Madoery†, Wenfeng Li*, Hong Yang§
*School of Electronic Science and Engineering, Nanjing University, Nanjing, China
† CONICET, Instituto de Estudios Avanzados en Ingeniería y Tecnología (IDIT). Córdoba, Argentina
‡Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
§Institute of Spacecraft System Engineering, China Academy of Space Technology (CAST), Beijing, China

***Abstract*—Next generation Global Navigation Satellite Systems (GNSSs) will embrace Inter-Satellite Links (ISLs) capability as an enabler of navigation and data transfer. However, platform restrictions will require of suitable Contact Plan Design (CPD) schemes which has been traditionally assumed *centralized*. After discussing the requirements of *distributed* CPD, we propose a first scheme in this class. Simulation results on the BeiDou GNSS prove that satisfactory metrics can be obtained enabling valuable and real autonomy for future GNSSs.**

***Index Terms*—Global Navigation Satellite Systems, Inter-Satellite Links, Contact Plan Design, Delay-Tolerant Networking**

I. Introduction

Over the past years, several Global Navigation Satellite Systems (GNSSs) were developed in different countries such as GPS in USA, GLONASS in Russia, BeiDou in China, Galileo in Europe, QZSS in Japan and IRNSS in India. As illustrated in Figure 1, the space segment of most GNSS networks comprises a set of several satellites in Medium Earth Orbit (MEO), Geostationary Orbit (GEO) and Inclined Geosynchronous orbit (IGSO) [1]. Present and next GNSSs generation are giving an increasingly important role to Inter-satellite links (ISLs). ISLs are already equipped in third-generation GPS satellites [2], are being considered in European [3], [4] and Chinese GNSSs [5], and have received significant attention from the research community [6]–[8]. Indeed, one of the main reasons for using ISLs lies in the fact that GNSS autonomy can be widely increased if the dependency on the ground segment is relaxed by meeting the following requirements:

1) ISL Ranging: to enable in-orbit clock synchronization, autonomous navigation and orbit determination.
2) ISL Data transfer: to relay telemetry and commands to and from satellites out of range from the ground station.

However, due to satellite platform restrictions, the number of ISL links that a satellite can establish at a given time is usually less than that of visible satellites. The reasons for this limitation depend on the specific satellite's antenna and transponder configuration, which can be classified as follows:

1) Single steerable high-gain antenna: a mechanically or electronically steered antenna is able to point to a single neighbor at a time.
2) Several fixed high-gain antennas: a sub-set of antennas placed in the satellite need to be chosen to accommodate:
   a) Limited on board power (which restricts simultaneous links).
   b) Limited available frequencies (i.e., interference).
   c) Limited transponders (i.e., more antennas than transponders connected via a switch or matrix).
3) Single fixed wide-beam antenna: a single antenna reaches many visible neighbors, but receiving neighbors might need to be scheduled to avoid collisions (negotiation-based medium access are discouraged at typical GNSS distances [9]).

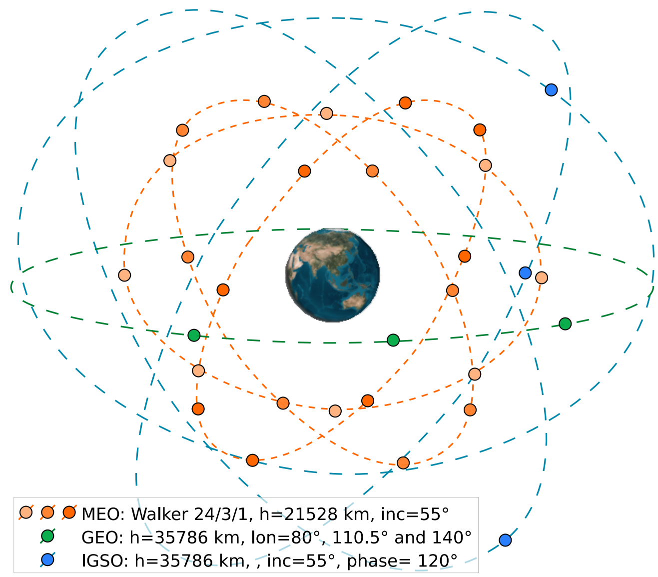


Fig. 1. BeiDou GNSS constellation and orbital parameters

Therefore, any of the latter cases (or a combination of them) demands a suitable ISL assignation schedule in time-evolving topologies, which results in a sporadic connected network. Solving such scheduling problem has been coined Contact Plan Design (CPD) [10]. In the particular case of GNSS, the CPD optimality criteria is dual as it must consider both frequent and diverse ranging measurements, and minimal end-to-end data delivery delay.

As surveyed in this paper, early ISL deployments in GPS, Galileo and BeiDou [2]–[5], as well as general CPD solutions for GNSS [11]–[28], have assumed that a central coordinator, usually a mission control center, is in charge of executing the CPD routine and timely provision the satellites with an appropriate contact plan. Based on this observation, we claim that a greater reduction in dependence on ground infrastructure can be achieved by decentralizing and distributing the contact plan calculation to satellites. This would enhance system autonomy, reduce operation costs and improve the system robustness under ground station outages. Furthermore, the fact that a common and accurate time-reference can be guaranteed among satellites in GNSS, makes of this application a perfect fit for distributed CPD. To the best knowledge of the authors, this is the first time that the distributed computation of contact plans is studied[1]. In particular, we discuss distributed CPD in the context of GNSS by considering its principles, benefits and requirements. Then, we propose and evaluate Distributed Fair Contact Plan (DFCP), the first appealing scheme based on a deterministic algorithm, which provides satisfactory results without the intervention of a centralized authority.

This paper is organized as follows. Section II provides surveys and overviews of centralized CPD schemes for GNSS. Then, Section III introduces distributed CPD in GNSS and discusses suitable computation strategies. Afterwards, calculation results based on the BeiDou GNSS system are analyzed in Section IV. Finally, conclusions are drawn in Section V.

II. Centralized CPD for GNSS

A. *Chronology of CPD with application to GNSS*

Although originally thought for low-orbit satellite systems, the general problem of link assignation in satellite networks can be directly applied to GNSSs operating in higher orbits. The first steps in the link assignation problem dates back to 1993 when Harathi et al. [11] and Noakes et al. [12] explored algorithms to optimize connectivity in highly partitioned networks. In 1998, Chang et al. extended the optimization objective to also model low-latency traffic flow volume [13]. In 2013, Huang et al. introduced the topology control for high-latency traffic in Delay-Tolerant Networks (DTN) [14]. In 2014, Fraire et al. continued this effort focused on satellite DTNs by considering fairness [15], routing [16] and traffic criteria [17] in the CPD problem.

However, in the context of navigation systems, ISL ranging requirements must also be included in the CPD optimization objective. Indeed, in 2011, Shi et al. described for the first time the cross-link optimization problem considering both ranging and communication needs [18]. In 2014, Han et al. discussed the ISL establishment criteria in a Walker-Delta GNSS constellation by using a theoretical background of the minimum of Position Dilution of Precision (PDOP) [19], which is also addressed in the present manuscript. In 2015, Li et al. analyzed a genetic algorithm for ground to GNSS satellite selection model based on Tracking, Telemetry and Control constraints [20]. In the same year, Yan et al. proposed a Simulated Annealing Design (SAD) to satisfy both requirements [21]. Yan's work is particularly interesting since data delivery was, for the first time, evaluated using well-known distributed DTN routing algorithms [30]. In 2018, Huang et al. studied a cascade optimization design which extended SAD by also optimizing the slot length to guarantee zero packet drop even in high traffic load [22]. Almost simultaneously, a theoretical model was proposed in [23] and authors in [24]–[26] explored similar heuristics to solve the CPD problem in GNSS, showing a notorious interest on the topic during the 2017-2018 period. Furthermore, works [27] and [28] provide additional evidence of China's commitment to the development of autonomous ISL-based GNSS constellations.

Table I summarizes the surveyed works and shows that the chronology of the CPD research with application to GNSS has ***(i)*** leaned towards a dual traffic and ranging objective to meet autonomy requirements in GNSS and ***(ii)*** intensified significantly in recent years with the advent of modern ISL-enabled GNSSs such as BeiDou in China.

TABLE I
Chronology of CPD with application in GNSS

Year	Requirement or objective		
	Traffic only	Ranging only	Traffic & Ranging
1993	[11], [12]		
1998	[13]		
2011			[18], [28]
2013	[14]		
2014	[15]	[19]	[27]
2015	[16]		[20], [21]
2016	[17]		
2017			[23], [24], [25]
2018			[22], [26]

B. *GNSS topology model and CPD criteria*

a) Topology model: Two navigation satellites can establish an ISL if ***(i)*** they are not shadowed by the earth, ***(ii)*** both ISL antennas lie within a given pointing angle and ***(iii)*** satellites distance is less than the maximum communication range. As satellites move and rotate along their orbital trajectory, ISL visibility changes and thus render a time-evolving topology. Resulting connectivity along a *scheduling interval* can be captured and divided by means of a series of Contact Plan Periods (CPPs) as illustrated for the two satellites in Figure 2. In general, CPP time lengths have been assumed homogeneous in order to facilitate and systematize network management, and kept in the order of 1 minute to preserve accuracy. Besides, CPPs can be further divided into smaller time slots in order to accomplish a finer granularity regarding ISLs assignation decisions. These CPD decisions are indeed necessary in GNSS in order to optimize *ranging* and *data delivery* criteria by taking into account satellite communication restrictions.

b) Ranging: A slot granularity, typically in the order of 3 seconds, allows to switch ISLs between many visible neighbors within a single CPP. The more diverse and numerous ISLs, the more precise the orbital determination for a source

[1]Notice that distributed CPD in [29] refer to the division of large multi-layer satellite networks contact plans in order to reduce computing delay. While such CPD occurs in a centralized node, this paper discusses CPD that can be executed and agreed between satellites without the support from a centralized node.

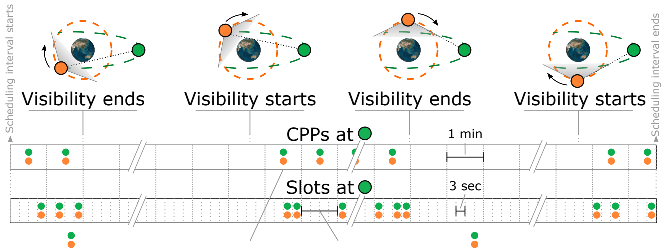


Fig. 2. GNSS topology model based on CPPs and slots

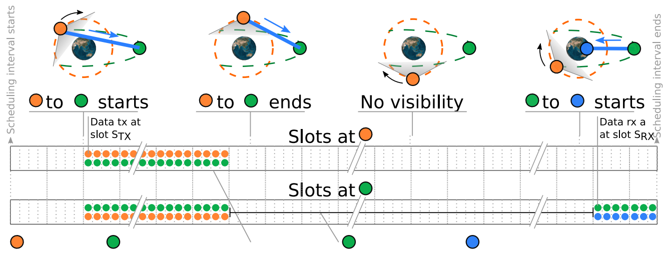


Fig. 3. GNSS store-carry-and-forward data flow

satellite. Specifically, the PDOP is a standard geometric measure to select good neighboring destination satellites in order to meet a desired positioning precision [31], [32]. As discussed in [26], the PDOP of satellite i at CPP k can be calculated by

$$PDOP_{i,k} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \quad (1)$$

where σ_x^2, σ_y^2, σ_z^2 are the principal diagonal elements of a matrix $Q_{i,k}$ defined as follows:

$$Q_{i,k} = (A_{i,k}^T A_{i,k})^{-1}. \quad (2)$$

Matrix A is composed of J rows whose elements are unit vectors (x_j, y_j, z_j) from source satellite i to each destination neighbor j that has been assigned a ISL at least once within CPP k. For example, if four ISLs are established with satellites $j1$ to $j4$ at CPP k, then $J = 4$ and

$$A_{i,k} = \begin{bmatrix} x_{j1} & y_{j1} & z_{j1} \\ x_{j2} & y_{j2} & z_{j2} \\ x_{j3} & y_{j3} & z_{j3} \\ x_{j4} & y_{j4} & z_{j4} \end{bmatrix}. \quad (3)$$

In general, the more satellites with the more diverse geometry, the less error and the lower the PDOP metric. It should be noticed that $J <= 1$ would always render an infinite PDOP as no position area in the 3D plane can be properly bounded from such conditions. Higher values of J might improve PDOP as long as measurements are geometrically diverse. In particular, a PDOP below 1 is considered ideal, from excellent to fair when in between 1 and 20, and poor when higher than 20. A system-level ranging metric can be conveniently obtained by averaging PDOP among all i satellites and k CPPs [19].

c) Data delivery: The application of CPD in GNSS typically results in a highly partitioned time-evolving topology without contemporaneous end-to-end connectivity. As a result, data might need to be temporarily stored in intermediate nodes, resulting in a store-carry-and-forward flow [33], [34]. Protocols based on the aforementioned DTN architecture [35], [36] are therefore a suitable standardized approach for automating data handling in ISL-enabled GNSS networks [21], [37], [38]. To this end, several centralized and distributed routing algorithms have been proposed, studied and implemented and successfully validated [30], [39]. As a simple and illustrative example, Figure 3 depicts how a telemetry transmitted at slot S_{TX} can be stored in an intermediate satellite before being delivered to its destination on ground at slot S_{RX}. The time period between S_{TX} and S_{RX} is considered the data *delivery delay* and is indeed dependent on the slot assignation and on the routing procedure executed by satellites. Total *transfered data volume* and *buffer utilization* are additional relevant metrics that depend on traffic and routing information (typically determined by simulation). Similar to ranging metrics, data delivery can also be conveniently averaged along time for all source-destination satellite pairs. As a result, efficient CPD schemes should assign ISLs by taking into account this type of data flow, a system-level data delivery metric, and without disregarding a ranging metric.

C. Centralized CPD approach

All the surveyed CPD efforts [11]–[28] have sought to optimize GNSS ranging and data delivery metrics by executing arbitrarily complex (heuristic) routines in a centralized mission control on Earth, where processing and memory capabilities are usually larger than in satellites. Once the CPP slots are assigned to satellite pairs, the contact plan is mapped to specific configuration frames which must be timely and individually distributed to each satellite to guarantee the continuous operation of the GNSS. Such frames must include all the necessary information for satellites to accurately point their electronically-steered antenna in the correct direction on each slot [22]. Therefore, when considering a constellation of dozens of satellites, where a scheduling interval may be comprised of several thousands of slots, configuration frames may result inconveniently large and challenging to provision. As a result, previous CPD schemes for GNSS have inadvertently forced the system to become highly dependent on a central entity, which goes against system autonomy, the original motivation to introduce ISLs in GNSSs in the first place [2]–[8].

III. Distributed CPD for GNSS

In this work we claim that enabling real GNSS autonomy via ISLs requires to embrace distributed CPD schemes. This approach requires satellites to separately compute their own contact plan without relying in a centralized entity. This independence regarding the ground station has multiple benefits:

1) Reduced operation costs: satellite operation complexity and thus costs can be reduced. Navigation systems can be conceived with fewer GNSS ground stations and less operation efforts in managing contact plan calculation, verification and provisioning.

2) Robustness: the system becomes tolerant to technical or sun outages [40], unplanned interference, jamming or potentially long unavailability periods of one, some or all GNSS ground stations.
3) Hybrid approach: as a centralized CPD scheme may yield better overall metrics due to its larger capability to handle more complex and computation-demanding problems, an hybrid approach could use this scheme in normal operation mode, while relying in distributed CPD as backup in case of unexpected problems on the ground segment. For example, if scheduled slots are not timely updated, satellites can react and calculate them based on the strategies discussed below.

To the best knowledge of the authors, and based on the brief survey presented in Section II-A, this is the first time that distributed contact plan calculation has been discussed in the context of space networking, and particularly for GNSS. However, implementing a distributed CPD has additional constraints and requirements that must be carefully considered.

A. Distributed CPD requirements

a) On-board topology determination: Satellites will need to propagate its own and its neighbors trajectories (and orientations) in order to determine the forthcoming network connectivity. In this regard, previous works have demonstrated the feasibility of running orbital propagation software on highly constrained on-board hardware while producing accurate ephemeris data [41]. Furthermore, the work presented in [42] discussed efficient on-board methods to enable local and remote orbit determination of GNSS Satellites. Although evidences show that meeting this requirement is technologically feasible, the specific details on the type and quality of information required on the satellites, as well as its correct determination based on ranging procedures, are out of scope of the present paper.

b) Scheduling interval synchronization: Scheduling intervals must be accurately synchronized in order to obtain the exact same schedule and implement it at exactly the same time in all GNSS satellites. Evidently, CPPs and slot times must also be homogeneously configured throughout the GNSS. The fact that on-board clocks are very accurate in GNSS satellites, with negligible drifts in the order of 9 to 18 ns per day [43], makes of GNSS a perfect candidate for distributed CPD. Besides, the start and duration of scheduling intervals, over which contact plans are calculated, can be conveniently agreed and fixed in advance. For example, 24 h-length schedules can be periodically calculated and implemented on each satellite at 00:00:00 UTC time.

c) Deterministic scheduling: In order to guarantee a correct GNSS operation, the derived schedule on each satellite must match with the decisions made in remote nodes. For example, the schedules of both satellites in Figure 3 need to be synchronized for one satellite to transmit data and the other to receive it on the same slots. In other words, both antennas should be simultaneously pointing to each other in order to successfully establish an ISL. As in [44], we classify these scheduling algorithms as *deterministic* since for a given input, they produce the same results following the same computation steps. Schemes not complying these conditions fall under the *randomized* classification. Because randomized algorithms typically throw coins during the execution, either the order of execution or the result of the algorithm might be different for each run on the same input. As a result, random heuristic strategies, which can provide efficient but different schedules on every execution, are not suitable for distributed CPD[2]. Since this is the case of most existing solutions revised in Section II-A, in this work we study deterministic CPD strategies for implementing distributed CPD solutions in future GNSS.

B. A distributed CPD scheme for GNSS

Among previous works surveyed in [11]–[28], only the Fair Contact Plan (FCP) scheme introduced by Fraire's et al. in [15] falls under the deterministic CPD classification. In this section, we describe the FCP formulation and adapt it to honor ranging and data delivery metrics in GNSS.

1) FCP overview: FCP exploits efficient matching algorithms in order to obtain contact plans for nodes that, despite having multiple possible neighbors, are capable of establishing only a single link at a time. Based on a dynamic programming approach, FCP sequentially iterates through successive slots in the forthcoming topology while deciding which pair of visible satellites will be scheduled with an ISL. As illustrated in Figure 4, the network connectivity on each slot $S_1, S_2 ... S_n$ is described by a graph $G(V, E, w)$ whose vertices V correspond to satellites and edges E (i.e., arcs) stand for the possibility of establishing an ISL between two satellites on such slot. Furthermore, each edge in FCP has a weight attribute w corresponding to the total accumulated time that such pair of nodes remained disconnected in previous iterations (i.e., without being assigned a scheduled ISL).

On a slot per slot basis, the resulting graph is subject to a *maximum weight non-perfect matching routine*. As discussed in detailed in the Appendix, such routine allows to determine a maximal selection of edges M such that the sum of w is maximal, and no other edge not in M can be added to M without breaking the matching condition [45]. In other words, from all the possible matchings M in a given slot, FCP chooses the one that provides the higher weight sum of all the arcs in M. As a result, at each slot, FCP schedules the set of ISLs that renders the higher sum of weights in the selection. Evidences in [15] show that resulting contact plans guarantee a fair (equal) distribution of total *contact time* between nodes along time.

In order to implement a maximum non-perfect matching routine, we consider a reduction of the maximum non-perfect

[2]It should be noted though, that if all satellite nodes are provisioned with the same seed used for the random number generation, then a heuristic could become deterministic in the sense that all satellites would be able to obtain the same schedule. Nevertheless, this would not solve the high processing effort required for these strategies to be carried out on satellite constrained computers and their exploration is left as future work.

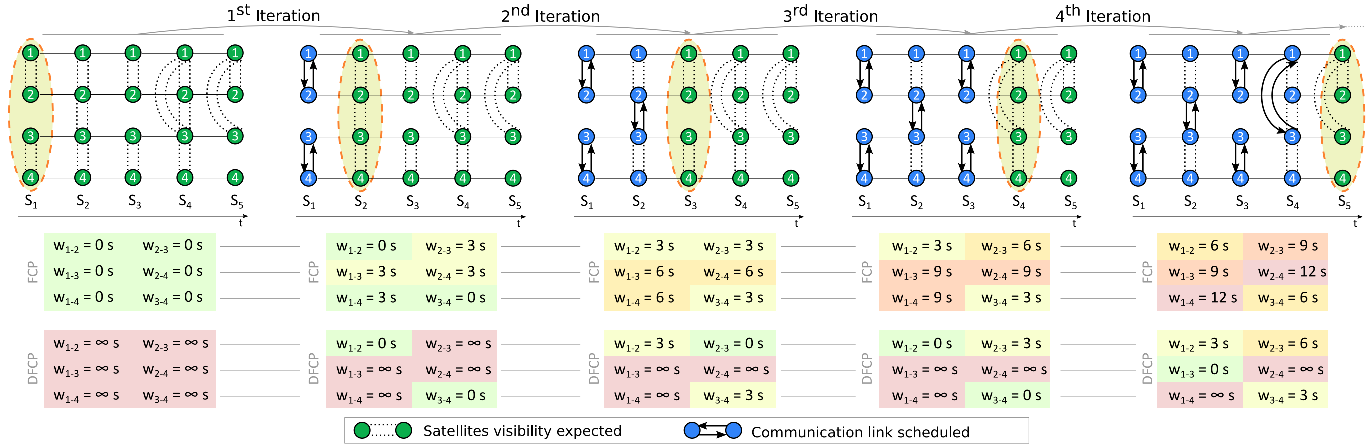


Fig. 4. A topology of four nodes whose time-evolving connectivity is expressed by a set of successive graphs corresponding to a series of five slots (S_1 to S_5). The weight of arcs from node i to and from j, depicted as $w_{i.j}$, are listed for both FCP and DFCP algorithms

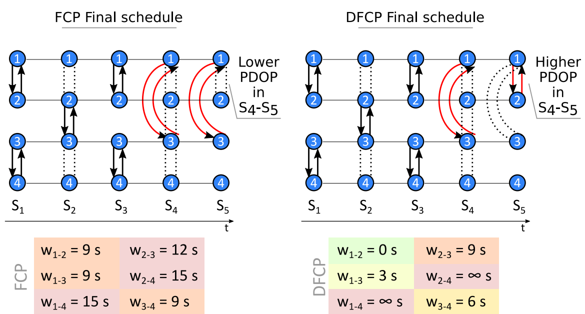


Fig. 5. Final iteration for FCP and DFCP provides schedules with different link assignation in state S_5 rendering different PDOP metrics

weighted matching to the perfect weighted matching proposed in [46]. The perfect weighted matching can then be solved by an efficient algorithm known as Blossom V [47], which is in turn based on the well known Blossom algorithm studied by Edmonds in [48]. Interested readers can refer to the Appendix for specific information on the implementation strategy chosen to solve the maximum non-perfect matching routine in FCP, and in the adaptations discussed below to use this scheme in GNSSs.

Optimizing contact time between nodes might derive in the scheduling of several contiguous slots to a single pair of satellites. For example, consider the 4th and 5th iterations (i.e., final FCP schedule) in Figure 4 and 5 respectively. After arc 1-3 is selected for slot S_4 at iteration 4, it is chosen again for slot S_5 at iteration 5. Supposedly, this in furtherance of equality in arc's weights and thus in node-to-node contact time. If possible, arcs with high accumulated weight will be scheduled by FCP for several successive slots until the combined weight for other arcs result higher. Indeed, the weights illustrated in Figure 5 are kept as equal as possible with the FCP scheme. While this favors a fair connectivity time in the long term (as originally sought in FCP), it might severely penalize ranging metrics such as PDOP, which requires a frequent switch of ISL neighbors within short periods of time.

2) DFCP overview: A trivial possibility to overcome FCP limitations when implemented in GNSS is to reset weights to 0 on a per-CPP basis. However, FCP will still provoke small occurrences of the aforementioned problem within a CPP. Although we refer and evaluate such variant as *reset FCP* or RFCP, we explore more definitive adaptations of the algorithm to improve GNSS metrics as follows. We propose *Distributed FCP* or DFCP, which is based on a new weighting strategy. Instead of weighting accumulated disconnection time between pair of nodes, DFCP weights the accumulated time since the ISL was scheduled for the last time. In this scheme, scheduled edges will weight zero in the immediate next slot. As exemplified in the weights of Figures 4 and 5, edges between visible nodes whose weights are very high (or infinite if never chosen) will be scheduled with very high chances in following iterations. In particular, in the 5th iteration, edges 1-2 (with weight 3 s) will be prioritized instead of 1-3 (with weight 0 s) for slot S_5. This improves the PDOP metric when measured along slots S_4 and S_5. Indeed, arcs which spent long periods of time without being scheduled are prioritized but without incurring in slot monopolization as in the original FCP. The hypothesis is that DFCP will enable both good PDOP and delivery delay metrics in GNSS as different nodes will be contacted on a highly frequent basis.

The formal behavior of DFCP is thus quite simple and is depicted in Algorithm 1. The procedure takes as input a three-dimensional visibility matrix $[Vm]$ comprised of S slots, each of which includes the connectivity information between all $N \times N$ nodes in the topology. A similar three-dimensional scheduling matrix $[Sm]$ is finally returned with the chosen ISL connectivity. Weight for each source-destination pair is captured in a list W of size $N \times N$, which is initialized to infinity in lines 1-2. Lines 3-4 iterate over each slot s to execute a maximum non perfect matching algorithm over the weighted graph $G(N, [Vm]_s, W)$. G is composed of N nodes (vertices) and edges in concordance with the visibility matrix $[Vm]_s$ at slot s. Each edge in G is affected by a weight attribute w. Resulting assignation at slot s is done by modifying the scheduling matrix $[Sm]$. Once the assignation is done for slot s, lines 5-9 update the edge weights in W for the next iteration as follows. If an edge with weight w, was elected in the slot, then $w = 0$; if not, w accumulates the duration of slot s for future iterations (it might continue being ∞).

Algorithm 1: DFCP

input : visibility matrix $[Vm]$ of size $S \times N \times N$,
output: scheduling matrix $[Sm]$ of size $S \times N \times N$,
1 **for** $w \in W$ **do**
2 | $w \leftarrow \infty$;
3 **for** $s \in S$ **do**
4 | $[Sm]_s \leftarrow$ `maxNPMatching` $(G(N, [Vm]_s,\mathrm{W}))$;
5 | **for** $w \in W$ **do**
6 | | **if** `edge` $(w) \in [Sm]_s$ **then**
7 | | | $w \leftarrow 0$;
8 | | **else**
9 | | | $w \leftarrow w+$ `duration` (s);

It is interesting to note, that although the loop starting at line 3 is bounded by the length S of the visibility matrix (i.e., number of slots), nothing impedes from breaking (or pausing) it earlier, use the obtained (shorter) scheduling interval and devote processing effort to other more urgent on-board activity. In other words, even though DFCP is a deterministic algorithm, it still benefits from preemptive features which are typical from heuristics such as simulated annealing or genetic algorithms. However, while stopping (or pausing) heuristic routines in advance might render a penalty on the quality of the result, DFCP output will only penalize the schedule length, meaning it will provide partial but well-calculated plans. Fortunately, as DFCP provides performance metrics on a slot-by-slot basis, partial plans can be applied immediately while the final calculation can be continued and delivered later. Such characteristic of DFCP provides unprecedented flexibility in the context of distributed CPD for GNSS, which requires a continuous and correct scheduling computed on resource-constrained on-board computers.

To wrap up, several reasons favor the utilization of the DFCP formulation as a distributed CPD scheme in GNSS: ***(i)*** it is suitable for the single electronically-steered antenna equipped in most GNSS satellites; ***(ii)*** its schedules comprise fairly distributed ISLs favoring ranging and delivery delay metrics; ***(iii)*** it allows to prioritize links by purposely increasing their weights (e.g., links with GEO can be preferred with respect to MEO); ***(iv)*** it is deterministic as each execution with the same topological input renders the same output schedule; ***(v)*** it is based on processing efficient matching algorithms favoring its implementation in on-board computers; ***(vi)*** its calculation effort grows linearly respect the quantity of slots (the required processing time for a specific schedule interval can be very well estimated in advance) and ***(vii)*** it can be executed dynamically (i.e., it can be stopped and resumed by-demand) or statically (i.e., all slots in the scheduling interval are calculated without interruption).

IV. Distributed CPD for GNSS Analysis

In this section we evaluate the performance of distributed CPD for GNSS. Specifically, we study the schedules produced by ***(i)*** the original FCP algorithm, ***(ii)*** the reset-based RFCP and ***(iii)*** DFCP. Furthermore, we provide metrics for ***(iv)*** the Simulated Annealing Design (SAD) presented in [21], which is used as a state-of-the-art benchmark for centralized CPD solutions. On the one hand, analyzing SAD results allows to estimate the performance loss of migrating from a centralized CPD to a more autonomous distributed CPD scheme. On the other hand, it enables to compare the processing effort, a mandatory analysis before considering any algorithm in a flight-grade on-board computer.

Similarly to [26], we consider the BeiDou GNSS which, once completely deployed in orbit, will comprise a ground station in Beijing, China, 24 MEO, 3 GEO and 3 IGSO satellites with the orbital parameters listed and illustrated in Figure 1. Since all BeiDou satellites in MEO, GEO and IGSO carry an electronically steerable antenna with bidirectional ISL capability to communicate with any node in the constellation, the system is particularly suitable for the discussed CPD procedures. Satellite positions were calculated for the scheduling interval using the High-Precision Orbit Propagator (HPOP) included in AGI's Systems Tool Kit (STK) software[3]. Further details on the simulation parameters are provided in Table II.

TABLE II
BeiDou scenario parameters

Scheduling interval	1,440 min (1 day)
CPP duration [25]	1 min (total of 1,440 CPPs)
Slot duration [25]	3 sec (total of 28,800 slots)
ISL pointing range in MEO	60°
ISL pointing range in GEO/IGSO	45°
Beijing GS pointing range	85°
Beijing GS coordinates	40.1172° lat, 116.228° long
ISL maximum range	Not set (∞)

A. Analysis

We analyze PDOP and data delivery delay metrics, by considering an all-to-all and all-to-ground traffic patterns. Furthermore, we discuss processing measurements in order to get some insights on the computing effort. Ordinates range of the curves are kept uniform when possible in order to facilitate comparison between schemes.

a) PDOP: The PDOP metric was averaged for all satellites within a CPP while registering the maximum and minimum values. Resulting curves are plotted in Figure 6. As expected, directly applying the original FCP to GNSS renders unacceptable results in terms of autonomous position determination. Actually, several measurements in FCP result

[3] Interested readers can access the BeiDou STK scenario files by means of the following URL: https://drive.google.com/file/d/1ncQW3zOB7n5GsepA3WLig9jNIqsNEsLq/view?usp=sharing

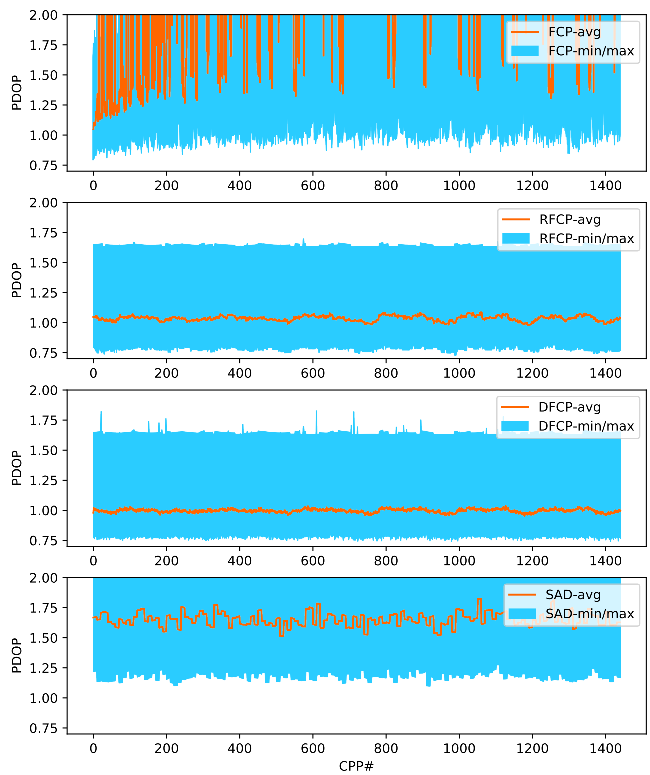


Fig. 6. PDOP metric averaged among all satellites along 1400 CPPs (1 day)

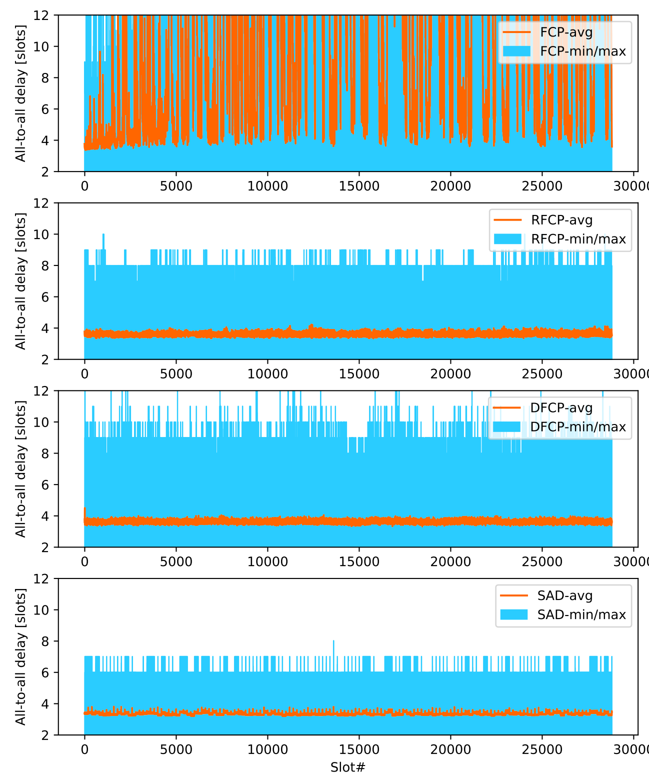


Fig. 7. All-to-all delivery delay averaged along 28800 slots (1 day)

in an infinite PDOP, which avoids to provide an average metric for the day period. On the other hand, the daily averaged PDOP for reset-based FCP (RFCP) is 1.03. The fact that RFCP exhibits a much better performance, is an evidence that the slot monopolization discussed in Section III is indeed the limitation of FCP in the context of ranging. This also explains the further improvement of DFCP, which provides a very suitable PDOP average of 0.99 in the 24 hs scheduling interval. These results show that DFCP is an appealing scheme in terms of precise autonomous navigation. It is interesting to note that DFCP also outperforms the state-of-the-art SAD scheme, which delivers an averaged PDOP of 1.65. The difference is such that even the best PDOPs among all satellites in SAD is not even close to the average PDOP obtained from DFCP. As we demonstrate with the data delivery delay results, the reason for the latter is that SAD scheme optimizes the delay while keeping PDOP under control only for a period of 8 slots, while DCFP considers 20 of them. [21].

b) Data delivery delay: The delivery delay was analyzed for two different traffic patterns: all-to-all and all-to-ground. In both cases, the delivery delay was calculated by using the Merugu's Floyd Warshall routing adaptation for store-carry-and-forward networks (i.e., DTN) [39]. Besides, in order to provide a continuous measurement of delay at the end of simulation, we have assumed that the connectivity of the network continues after the last slot (slot 28800) to the first (as if the scheduling interval were periodic).

On the one hand, an all-to-all traffic pattern allows to measure the behavior of the system when traffic data is homogeneously transmitted among nodes. This traffic metric is illustrated in Figure 7. As with the PDOP, FCP provides the worst averaged all-to-all delay of 19.67 slots (i.e., 59.02 sec). RFCP, with an average delay of 3.62 slots (i.e., 10.87 sec) is slightly better than DFCP with a delay of 3.64 slots (i.e., 10.94 sec). The reason for the latter can be analyzed from Table III, which summarizes how many ISLs have been scheduled per slot on each scheme. Given the restriction of one ISL per satellite at the same time, the maximum possible ISLs per slot that can be enabled on the scheduling interval is 432000. RACP achieves such slightly better connectivity, but at the expense of an unequal distribution of ISLs which severely penalizes PDOP. SAD is the best in terms of all-to-all delivery delay, which is precisely the optimization goal of its objective function. Specifically, SAD provides an average delay of 3.35 slots (10.06 sec). Although the averaged delay of SAD was very close to RFCP and FCP, the worst case delay in SAD is never higher than 7 slots (21 sec), while in RFCP and DFCP is between 10 and 12 slots. In other words, SAD makes a good work at minimizing the worst case delays.

On the other hand, in order to mimic telemetry download, data delivery delay is measured from all satellites to the BeiDou ground station in Beijing. These measurements are plotted in Figure 8. Even though none of the studied schemes was designed to honor an all-to-ground delay, it is still an important metric to observe and control in GNSS operations.

TABLE III
SCHEDULED ISL PER SLOTS ALONG THE SCHEDULING INTERVAL

Scheme	Scheduled ISLs per slots
FCP	431999
RFCP	432000
DFCP	417124
SAD	432000

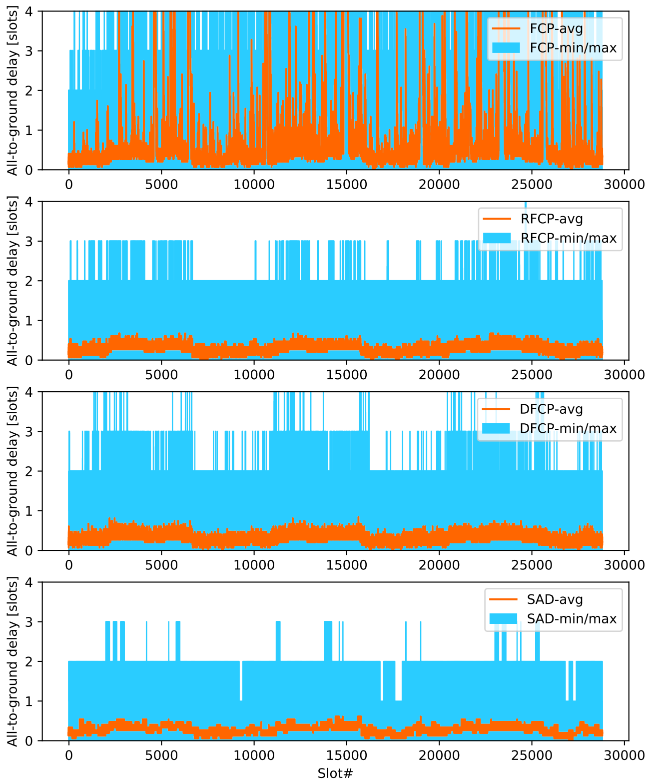


Fig. 8. All-to-ground delivery delay averaged along 28800 slots (1 day)

Illustrated in Figure 8, results show that delays to ground station are way below the measured in the all-to-all traffic pattern. In particular, FCP is the worst performing with an average delay of 1.03 slots (3.11 sec). Similarly with the all-to-all delay effect, RFCP shows an average delay of 0.28 slots (0.85 sec) which is slightly better than DFCP with an average delay of 0.31 slots (0.95 sec). The same reason holds: RFCP connectivity is better, but at the cost of severe slot monopolization. SAD is practically equal to RFCP, being the two the most performant in terms of data delivery delay to the ground station in Beijing.

As a final performance comparison of all evaluated schemes, we provide a scatter diagram in Figure 9. Being PDOP and all-to-all delay, the abscissas and ordinates axis respectively, the closer the scatter to the origin, the better for the GNSS. It can be seen that FCP is highly heterogeneous regarding its outputs, many of which are left out of the graph to focus on the most performant solutions. Because delivery delay is within its main optimization goals, SAD provides the best delay metrics and moderate PDOPs in comparison with RFCP and DFCP.

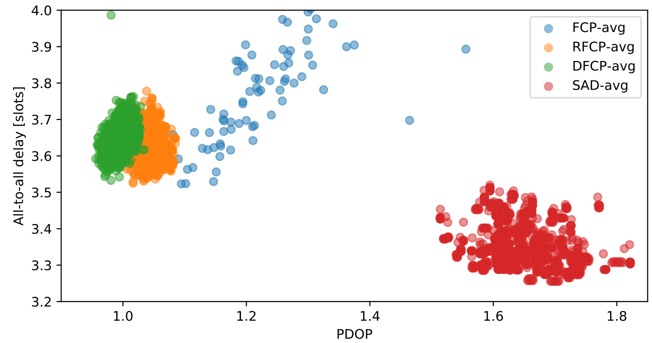


Fig. 9. Scatter graph for average delay and PDOP values of all schemes

SAD also shows a better output concentration (i.e., stability), but the average forms of RFCP and DFCP are even more stable along slots. It is noticeable in this plot that the PDOP improvement in DFCP can be much more important than the slight advantage in delay metric from RFCP.

Indeed, DFCP is slightly better than other schemes in terms of PDOP, and quite close in terms of average delay metric. However, the point of this performance comparison is not to show that DFCP outperforms SAD or RFCP, but to demonstrate that DFCP solution is the first of its kind that can be executed in a distributed fashion while still providing schedules of similar quality than traditional centralized CPD schemes for GNSS.

c) Computation effort: Calculation time was measured for the scenario under analysis. An Intel(R) Core(TM) i5-4570 CPU clocking at 3.20 GHz with a system memory of 8.00 GB running a Windows 10 operating system was used as a benchmarking platform. Table IV lists the computation effort measurements in terms of computational time (i.e., the length of time required to perform the complete computational process) and on a ratio between the observed Computational Time to Scheduled Time (CT/ST), in this case, 1440 minutes. In particular, CT/ST is an appealing indicator to depict what fraction of the time the processors need to invest in order to guarantee a continuous operation of the GNSS. Evidently, the slower the processor, the higher the CT/ST ratio. Results show that, by devoting 1.3% of the processing power of the test-bench processor is enough for DFCP, while a 30% is required if the node executes SAD. The main reason for the drastic reduction in compute time of DFCP is that it is a deterministic routine (i.e., based on Blossom algorithm), while SAD is an heuristic based on trial and error approaches that requires significant amount of potentially unusable calculations. These performance analysis validates the hypothesis that the processing demand of DFCP is much more convenient for resource-constrained on-board computers than heuristic approaches. In particular, to be able to accommodate the computational load of DFCP, satellites will need to allocate the equivalent processing power of roughly 1.3% of a standard desktop CPU. Although accurate processing effort estimations depends on the interaction between the final software implementation and the flight hardware characteristics, previous works have shown that on average, space-grade CPUs are typically 25-50x slower than desktop CPUs [49], indicating that running DFCP in a flight on-board computer is feasible.

TABLE IV
COMPUTATION EFFORT MEASUREMENTS

	DFCP	SAD
Computational time	1201.1 sec (20 min, 01 sec)	30558.9 sec (08 hs, 29 min, 18 sec)
CT/ST ratio	0.01390	0.3536

To wrap up this section, the presented analysis proved that: ***(i)*** heuristic-based approaches like SAD might provide good results for centralized CPD but at the expense of high processing effort, ***(ii)*** FCP overcomes the processing limitation and allows for distributed CPD but it is not suitable for GNSS as originally formulated, ***(iii)*** DFCP and RFCP solve the latter problems and provide very efficient PDOP and data delay metrics, but combined delay and navigation results suggest that DFCP is, at the moment, an overcoming scheme to tackle the distributed CPD problem in ISL-based GNSS.

B. Outlook

As the first in proposing distributed CPD for GNSS, this work opens a new research area. Among interesting future research opportunities in this domain we highlight: ***(i)*** the optimization of the arc weighting strategy in DFCP in order to improve PDOP if selected; ***(ii)*** the more precise determination of performance loss against other heuristics such as [22]; ***(iii)*** the application of new algorithms for distributed CPD in GNSS with multiple simultaneous ISL capability (DFCP assumes only one ISL is chosen at any satellite at any given moment); ***(iv)*** the analysis of tolerance against unexpected failures or intentional attacks; and ***(v)*** the discovery of new algorithms that could improve the processing performance (i.e., CT/CS indicator) of DFCP. About the latter point, in DFCP all satellites calculate the same global schedule based on the same input data (trajectory propagations are assumed identical). While this implies that the distribution strategy of DFCP is routine replication, there might be room for improvement if each satellite could compute and schedule only those slots where it participates as a sender or receiver. However, how to guarantee homogeneous decisions among all GNSS nodes in such approach is an appealing future research challenge that remains to be met.

V. CONCLUSION

In this work, we introduced the concept of distributed CPD for GNSS with ISL capability. Based on a brief survey of previous CPD solutions, we discovered that they have forced GNSS systems to become highly dependent on a central entity, which is against the original motivation of introducing ISLs. Indeed, a centralized scheduling is a reasonable approach in early deployments of ISLs in GPS, Galileo and BeiDou as it allows to keep a close control and gain confidence on novel ISL technology. Nonetheless, in this paper, we claimed that a distributed and automated scheduling approach will naturally take over in the long term. After discussing the advantages supporting the latter statement, we described the requirements for distributed CPD and proposed a suitable scheme coined Distributed Fair Contact Plan or DFCP. Evaluation results proved that DFCP can provide reasonable metrics in comparison with state-of-the-art centralized schemes while being the first scheme enabling a distributed scheduling and real autonomy for distributed navigation and data transfer in future GNSSs.

APPENDIX
MAXIMUM NON-PERFECT MATCHING IMPLEMENTATION

This appendix provides further details on how we obtained the maximum non-perfect matching in the evaluated FCP and DFCP algorithms. To this end we go over the concepts of matching, weighted matchings and perfect weighted matchings to arrive to non-perfect weighted matchings which is the correct routine to implement FCP and DFCP CPDs.

a) Matching: By definition, a *matching* is a set of edges without common vertices in a given graph. Although bipartite matchings can be easily computed by network flow algorithms, the problem in general graphs requires of further attention. Given a $G(V,E)$, a matching M in G is a set of non-adjacent edges in such a way that no two edges share a common vertex. A matching M is a *maximal matching* if any edge not in M is added to M it is no longer a matching. A maximal matching is a *maximum matching* if it contains the largest possible number of edges, meaning that $|M|$ is maximized. A maximum matching is also known as maximum-cardinality matching and is the problem that Edmonds back in solved in polynomial amount of computation time by means of the so called Blossom algorithm [48].

b) Weighted matching: However, in the context of DFCP, we are not interested in cardinal matchings but on *weighted matchings*. In particular, weights w assigned to edges E in $G(V,E,w)$ should prioritize arcs with higher weight while still producing a matching in order to take advantage of the higher number of ISL opportunities. Specifically, we seek for a set M that produces a matching of maximum (or minimum) total weight. As discussed in [45], the weighted matching problem can be solved by a combinatorial algorithms based on the cardinal Edmonds' algorithm as a subroutine. Blossom V is an efficient implementation of one of these formulations and is chosen for DFCP.

c) Perfect weighted matching: Blossom V computes a *perfect weighted matching of minimum cost* for which a C++ code is publicly available [47]. A matching M is perfect if it matches all vertices of the graph covered by M. It is also known as 1-factor, and complete matching and the quantity of vertex in G ($|V|$) must be even. In the graph of slot S_2, in Figure 4, the Blossom V algorithm will always choose the perfect matching M comprised of edges $e_{1,2}$ and $e_{3,4}$, disregarding the weight $e_{2,3}$. However, DFCP would need to be able to honor $e_{2,3}$ if the combined weight of $e_{1,2}$ and $e_{3,4}$

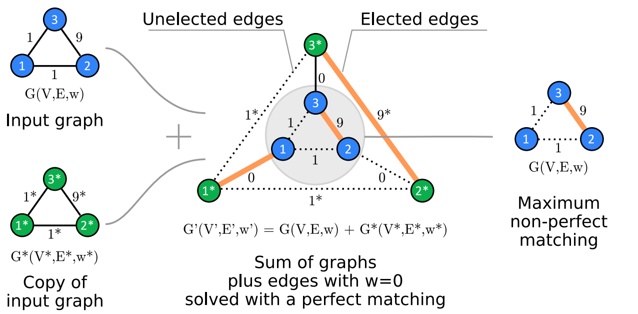


Fig. 10. Guido Schäfer's reduction of the maximum non-perfect matching to a perfect matching problem

is lower, and should also be able to accommodate graphs with odd vertices.

d) Non-perfect weighted matching: To generalize the applicability of Blossom V, Guido Schäfer in his Master's thesis proposes a reduction of the maximum non-perfect matching to a perfect matching problem [46]. The reduction doubles the size of the graph by applying the following procedure which is also illustrated with an example in Figure 10:

1) Let our graph be $G(V, E, w)$, we need a transformation $T(G) = G'(V', E', w')$ as follows:
2) Let $G^*(V^*, E^*, w^*)$ be a copy of G such that:
   a) every vertex $v^* = v$
   b) every edge $e^* = e$
   c) every weight $w^* = w$
3) Now consider the sum $G' = G + G^*$ plus edges from all v^* to v with 0 weight. This is:
   a) $V' = V \cup V^*$
   b) $E' = E \cup E^* \cup \{vv^* : v \text{ is in } V \text{ and } v^* \text{ is in } V^*\}$
   c) $w' = \{w$ when in E, w^* when in E^*, and 0 when $vv^*\}$
4) Now, if we solve the maximum perfect matching in G', we will have a maximum non-perfect matching in G.

Specifically, we implemented Schäfer's reduction in C++ and solve the maximum perfect matching in G' by using Blossom V routine which is available in the same language[4]. The resulting graph G contains the edges that corresponds to the ISLs that should be scheduled on the slot where the maximum non-perfect matching routine was applied.

REFERENCES

[1] G. X. Gao and P. Enge, "How many gnss satellites are too many?" *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2865–2874, October 2012.

[2] O. Luba, L. Boyd, A. Gower, and J. Crum, "Gps iii system operations concepts," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 1, pp. 10–18, Jan 2005.

[3] M. Sánchez, J. Pulido, F. Amarillo, and J. Gerner, "The esa gnss+ project. inter-satellite ranging and communication links in the frame of the gnss infrastructure evolutions," *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008), Savannah, GA*, pp. 2538 – 2546, 2008.

[4]Interested readers can access the C++ code files used in this work by means of the following URL: https://drive.google.com/file/d/1AVNSVIEnVAH5jVYZHQeJ8kCk5rop2Gws/view?usp=sharing

[4] F. A. Fernández, "Inter-satellite ranging and inter-satellite communication links for enhancing gnss satellite broadcast navigation data," *Advances in Space Research*, vol. 47, no. 5, pp. 786 – 801, 2011, scientific applications of Galileo and other Global Navigation Satellite Systems - II.

[5] J. Liu, T. Geng, and Q. Zhao, "Enhancing precise orbit determination of compass with inter-satellite observations," *Survey Review*, vol. 43, no. 322, pp. 333–342, 2011.

[6] J. Huang, Y. Su, W. Liu, and F. Wang, "Adaptive modulation and coding techniques for global navigation satellite system inter-satellite communication based on the channel condition," *IET Communications*, vol. 10, no. 16, pp. 2091–2095, 2016.

[7] "Establishment criteria, routing algorithms and probability of use of inter-satellite links in mixed navigation constellations," *Advances in Space Research*, vol. 51, no. 11, pp. 2084 – 2092, 2013.

[8] F. Dong, J. Lv, Y. Yu, Q. Wang, and C. Wang, "Inter-satellite traffic data modeling for gnss," in *China Satellite Navigation Conference (CSNC) 2012 Proceedings*, J. Sun, J. Liu, Y. Yang, and S. Fan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 69–78.

[9] J. A. Fraire, P. G. Madoery, J. M. Finochietto, P. A. Ferreyra, and R. Velazco, "Internetworking approaches towards along-track segmented satellite architectures," in *2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, Sept 2016, pp. 123–128.

[10] J. A. Fraire and J. M. Finochietto, "Design challenges in contact plans for disruption-tolerant satellite networks," *Communications Magazine, IEEE*, vol. 53, no. 5, pp. 163–169, May 2015.

[11] K. Harathi, P. Krishna, R. E. Newman-Wolfe, and R. Y. C. Chow, "A fast link assignment algorithm for satellite communication networks," in *Proceedings of Phoenix Conference on Computers and Communications*, Mar 1993, pp. 401–408.

[12] M. D. Noakes, J. B. Cain, J. W. Nieto, and E. L. Althouse, "An adaptive link assignment algorithm for dynamically changing topologies," *IEEE Transactions on Communications*, vol. 41, no. 5, pp. 694–706, May 1993.

[13] H. S. Chang, B. W. Kim, C. G. Lee, S. L. Min, Y. Choi, H. S. Yang, D. N. Kim, and C. S. Kim, "Fsa-based link assignment and routing in low-earth orbit satellite networks," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 3, pp. 1037–1048, Aug 1998.

[14] M. Huang, S. Chen, Y. Zhu, and Y. Wang, "Topology control for time-evolving and predictable delay-tolerant networks," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2308–2321, Nov 2013.

[15] J. A. Fraire, P. G. Madoery, and J. M. Finochietto, "On the design and analysis of fair contact plans in predictable delay-tolerant networks," *Sensors Journal, IEEE*, vol. 14, no. 11, pp. 3874–3882, Nov 2014.

[16] J. A. Fraire and J. Finochietto, "Routing-aware fair contact plan design for predictable delay tolerant networks," *Ad Hoc Networks*, vol. 25, pp. 303 – 313, 2015.

[17] J. A. Fraire, P. G. Madoery, and J. M. Finochietto, "Traffic-aware contact plan design for disruption-tolerant space sensor networks," *Ad Hoc Networks*, vol. 47, pp. 41 – 52, 2016.

[18] L. y. Shi, W. Xiang, and X. m. Tang, "A link assignment algorithm for gnss with crosslink ranging," in *2011 International Conference on Localization and GNSS (ICL-GNSS)*, June 2011, pp. 13–18.

[19] S. Han, Q. Gui, G. Li, and Y. Du, "Minimum of pdop and its applications in inter-satellite links (isl) establishment of walker constellation," *Advances in Space Research*, vol. 54, no. 4, pp. 726 – 733, 2014.

[20] L. Jing, Z. Tianjiao, and Y. Gangqiang, "Satellite-ground tt&c united scheduling methods of gnss constellation based on nodes constraint," in *China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume I*, J. Sun, J. Liu, S. Fan, and X. Lu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 55–66.

[21] H. Yan, Q. Zhang, Y. Sun, and J. Guo, "Contact plan design for navigation satellite network based on simulated annealing," in *2015 IEEE International Conference on Communication Software and Networks (ICCSN)*, June 2015, pp. 12–16.

[22] J. Huang, W. Liu, Y. Su, and F. Wang, "Cascade optimization design of inter-satellite link enhanced with adaptability in future gnss satellite networks," *GPS Solutions*, vol. 22, no. 2, p. 44, Feb 2018.

[23] C. Xiaogeng and C. Yuning, "Time division inter-satellite link topology generation problem: Modeling and solution," *International Journal of Satellite Communications and Networking*, vol. 36, no. 2, pp. 194–206, 2017.

[24] J. Huang, Y. Su, W. Liu, and F. Wang, "Optimization design of inter-satellite link (isl) assignment parameters in gnss based on genetic algorithm," *Advances in Space Research*, vol. 60, no. 12, pp. 2574 – 2580, 2017, bDS/GNSS+: Recent Progress and New Applications - Part 2.

[25] D. Yang, J. Yang, and P. Xu, "Timeslot scheduling of inter-satellite links based on a system of a narrow beam with time division," *GPS Solutions*, vol. 21, no. 3, pp. 999–1011, Jul 2017.

[26] L. Sun, Y. Wang, W. Huang, J. Yang, Y. Zhou, and D. Yang, "Inter-satellite communication and ranging link assignment for navigation satellite systems," *GPS Solutions*, vol. 22, no. 2, p. 38, Jan 2018.

[27] D. Wang, B. Xu, W. Liu, and G. Sun, "A novel navigation inter-satellite links ranging hierarchy and its orbit determination performance," in *Guofang Keji Daxue Xuebao/Journal of National University of Defense Technology*, vol. 36, 02 2014, pp. 62–66.

[28] J. Lin, J. Yang, and Y. Wang, "Optimizing algorithm for the scheduling of gnss crosslink: a matrix transform approach," in *The 2nd China Satellite Navigation Conference, Shanghai, China*, 2011, p. 467–481.

[29] W. Shi, D. Gao, H. Zhou, B. Feng, H. Li, G. Li, and W. Quan, "Distributed contact plan design for multi-layer satellite-terrestrial network," *China Communications*, vol. 15, no. 1, pp. 23–34, Jan 2018.

[30] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, "Contact graph routing in dtn space networks: overview, enhancements and performance," *IEEE Comms. Magazine*, vol. 53, no. 3, pp. 38–46, March 2015.

[31] R. B. Langley, "Dilution of precision," *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.

[32] M. Barbarella, S. Gandolfi, L. Poluzzi, and L. Tavasci, "Precision of ppp as a function of the observing-session duration," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2018.

[33] K. Zhao, R. Wang, S. C. Burleigh, A. Sabbagh, W. Wu, and M. D. Sanctis, "Performance of bundle protocol for deep-space communications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2347–2361, October 2016.

[34] G. Yang, R. Wang, A. Sabbagh, K. Zhao, and X. Zhang, "Modeling optimal retransmission timeout interval for bundle protocol," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2018.

[35] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture," Internet Requests for Comments, RFC Editor, RFC 4838, April 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4838.txt

[36] K. Scott and S. Burleigh, "Bundle protocol specification," Internet Requests for Comments, RFC Editor, RFC 5050, November 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5050.txt

[37] R. Sorace, "Overview of multiple satellite communication networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 4, pp. 1362–1368, Oct 1999.

[38] M. Mohorcic, A. Svigelj, and G. Kandus, "Traffic class dependent routing in isl networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, no. 4, pp. 1160–1172, Oct 2004.

[39] S. Merugu, M. Ammar, and Z. E., "Routing in space and time in networks with predictable mobility," Georgia Institute of Technology, Tech. Rep., January 2004.

[40] X. T. Vuong and R. J. Forsey, "Prediction of sun transit outages in an operational communication satellite system," *IEEE Transactions on Broadcasting*, vol. BC-29, no. 4, pp. 121–126, Dec 1983.

[41] D. R. Hartmann, "Power savings through onboard orbit propagation for small satellites like npsat1," 2016, phD. Thesis, Naval Postgraduate School, Monterey, California.

[42] H. Wang, Z. Chen, J. Zheng, and H. Chu, "A new algorithm for onboard autonomous orbit determination of navigation satellites," *Journal of Navigation*, vol. 64, no. S1, p. S162–S179, 2011.

[43] D. Sathyamoorthy, S. Shafii, Z. F. M. Amin, A. Jusoh, and S. Z. Ali, "Evaluating the effect of global positioning system (gps) satellite clock error via gps simulation," *IOP Conference Series: Earth and Environmental Science*, vol. 37, no. 1, p. 012013, 2016.

[44] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms, 3rd Edition*. Cambridge, MA: MIT Press, 2009.

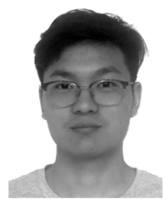
[45] W. W., "Lovász, l.; plummer, m. d., matching theory. budapest, akadémiai kiadó 1986. xxxiii, 544 s., ft 680,—. isbn 9630541688," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 68, no. 3, pp. 146–146.

[46] G. Schäfer, "Weighted matchings in general graphs," Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, 2000.

[47] V. Kolmogorov, "Blossom v: a new implementation of a minimum cost perfect matching algorithm," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 43–67, Jul 2009.

[48] J. Edmonds, "Path, trees, and flowers," *Can. J. Math*, vol. 17, pp. 449–467, 1965.

[49] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, and G. Furano, "High-performance embedded computing in space: Evaluation of platforms for vision-based navigation," *Journal of Aerospace Information Systems*, vol. 15, no. 4, pp. 178–192, 2018.



Zhibo Yan received his B.S. degree from Nanjing University, Nanjing, China, in 2017. He is currently a PhD candidate in School of Electronic Science and Engineering and the Institute of Space-Terrestrial Intelligent Networks at Nanjing University. His research interests include space information networking, satellite communications and networking.

Juan A. Fraire is an assistant researcher at the National Research Council of Argentina (CONICET) and a associate professor at FAMAF, Universidad Nacional de Córdoba (UNC), University of Saarland in Germany and guest professor at Politecnico di Torino in Italy. His research focuses on spaceborne networking and applications. Juan is the funder and chair of the annual Space-Terrestrial Internetworking Workshop (STINT) since 2014, and has co-authored more than 45 papers published in international journals and leading conferences.

Kanglian Zhao (M'11) received the B.Sc. and Ph.D. degrees from Nanjing University (China), in 2003 and 2014 respectively. He was with IMEC and K. U. Leuven as an international scholar from Aug. 2009 to Aug. 2010. He is now an associate professor in School of Electronic Science and Engineering and the Institute of Space-Terrestrial Intelligent Networks, Nanjing University. His current research interests include network architecture and emulation for space internetworking, satellite-terrestrial integrated communications.

Hongcheng Yan received the B.S. degree in electronic science and technology and M.S. degree in instrument science and technology, in 2008 and 2011, respectively, both from North University of China, Shanxi, China. He received the Ph.D. degree in flight vehicle design from China Academy of Space Technology, Beijing, China, in 2015. He is now a senior engineer in Beijing Institute of Spacecraft System Engineering, Beijing, China. His current research interests include network architecture and protocol design for satellite-terrestrial integrated networks.

Pablo G. Madoery received a Telecommunications Engineering degree at the Instituto Universitario Aeronáutico and a PhD in Engineering Sciences at the National University of Córdoba (UNC). He has been involved in several national and international research projects in the field of satellite communication networks, and is currently studying delay and disruption tolerant networks as part of his postdoctoral research.

Wenfeng Li (M'08) received his Ph.D. degree at National Mobile Communications Research Laboratory, Southeast University, China, in 2012. From August 2012 to April 2017, he was a Postdoctoral Fellow at the School of Electronic Science and Engineering, Nanjing University, China. In April 2017, he joined Nanjing University, where he is an Associate Research Fellow. His current research interests include space network architecture, satellite communications and networks, wireless and mobile communications.

Hong Yang received his Bachelor and Master degree from Northwest Telecommunication Engineering College and China Academy of Space Technology (CAST) in 1984 and 1991 respectively. He studied in International Space University, NASA AMES Center during Jul. - Aug. 2009. He has been working in CAST since 1991, where he is currently a professor in the Institue of Manned Spce System. He is also an adjunct professor in Nanjing Unviersity. Professor Yang worked as designer and deputy chief designer of Shenzhou series manned spaceships and chief designer of Tiangong-1 vehicle. He has been as the chief designer of Chinese space station system since 2011. Professor Yang is a Fellow of International Academy of Astronautics. He has won various awards for his achievement in space vehicle designing, inculding top grade China National Scientific and Technological Prizes for two times in 2004 and 2013 respectively. His recent research interests include information engineering, system engineering and space vehicle designing.