2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)

Automated Data Retrieval from Large-Scale Distributed Satellite Systems

Dominik Krupke[1], Volker Schaus[2], Andreas Haas[1], Michael Perk[1], Jonas Dippel[1] Benjamin Grzesik[2], Mohamed Khalil Ben Larbi[2], Enrico Stoll[2], Tom Haylock[3], Harald Konstanski[3], Kattia Flores Pozo[3], Mirue Choi[3], Christian Schurig[4], and Sándor P. Fekete[1]

Abstract— **We consider a central challenge that is mission critical for the successful operation of large-scale satellite constellations in Low-Earth Orbit: How can we coordinate the short-term download operations for the enormous amounts of generated data, based on wireless line-of-sight connections to a limited number of stationary ground station? These issues are critical for the future growth of space systems, with multiple commercial space operators competing for downloading their commercial data in a timely fashion, relying on the services of a scarce set of ground stations that is subject to numerous strong constraints, so it cannot simply be expanded.**

We present a distributed auction-based scheduling approach for maximizing the value of the downloaded data. Our method allows competing satellite operators to bid for contact times and has a fair and transparent price estimation based on the competition. On its own, it can also be used with a simple bidding strategy to obtain good schedules; this is demonstrated on benchmark simulation with up to 1080 satellites. As a consequence, we are able to achieve values rates of 74% of available data, compared to 28% for standard greedy strategies.

I. INTRODUCTION

The trend of using distributed space systems–such as satellite constellations–instead of monolithic systems has been growing for the last decade. Some of the newly announced constellations feature more than 1000 satellites. Traditional spacecraft operations involve manual control of the spacecraft by skilled human operators, following a 4-eyes principle. Even when operators batch multiple telecommands together, this process is still time-consuming and prone to errors. In addition, the autonomy level aboard traditional spacecraft is low; critical activities, such as anomaly identification and resolution, must be supervised by the operators. This manual approach is not (linearly) scalable to large satellite constellations, resulting in a variety of algorithmic challenges for automated operation. According to the NASA Technology Roadmap [1]: “Demand continues for ground systems which will plan more spacecraft activities with fewer commanding errors, provide scientists and engineers with more functionality, process and manage larger and more complex data more quickly, all while requiring fewer people to develop, deploy, operate and maintain them”. This gives rise to numerous problems of robotics and automation, many of which involve extremely complex systems and requirements.

One of the critical aspects of satellite constellation operation is being able to promptly retrieve the generated payload data. For satellites in low-earth orbit (LEO), this data is typically based on visual imaging, with total size growing quadratically with increasing image resolution and linearly with time resolution; in fact, the demand for up-to-date, fine-grained images is one of the driving forces behind the rapidly expanding number of satellite in current and projected constellations. Downloading this data relies on wireless communication between the orbiting satellites and a limited number of stationary ground stations with powerful parabolic antennas, requiring a direct line of sight; note that even when the underlying metadata (describing size and importance of payload data consisting of image files) is relatively small (and thus rapidly communicated), the download times for the actual payload data itself are considerable. As a consequence, these communication links become a critical bottleneck for successful, timely operation of a satellite constellation. However, coordinating these communications for massive satellite constellations and enormous amounts of data involves excruciatingly difficult optimization problems.

Currently, most satellite systems have their own dedicated ground stations. However, the demand for growing the number of satellites (e.g., for providing high-resolution, short-term imaging data) is quite high, whereas the building and operation of ground station (which should be located at high geographic latitudes, i.e., close to the poles in order to provide frequent communication windows) is subject to numerous geographic, logistic and political constraints, as well as quite expensive. As a consequence, it is foreseeable that ground stations will have to be shared, enabling these systems to be more dynamic and deal with temporary bandwidth bottlenecks while also lowering the prerequisites of

*This work was supported by the European Space Agency, project ASIMOV (“Automated conStellatIon Management Of space Vehicles”).

[1]Dominik Krupke, Andreas Haas, Michael Perk, Jonas Dippel, and Sándor Fekete are with the Department of Computer Science, TU Braunschweig, Germany. {krupke,haas,mperk,jodippel,fekete} @ibr.cs.tu-bs.de

[2]Volker Schaus, Benjamin Grzesik, Mohamed Khalil Ben Larbi, and Enrico Stoll are with the Institute of Space Systems, TU Braunschweig, Germany. {v.schaus,b.grzesik,m.ben-larbi,e.stoll} @tu-bs.de

[3]Tom Haylock, Harald Konstanski, Kattia Flores Pozo, and Mirue Choi are with Planet Labs, Berlin, Germany. {tom.haylock,harald.konstanski,kattia.flores, mirue.choi}@planet.com

[4]Christian Schurig is with the European Space Operations Centre, European Space Agency, Darmstadt, Germany. Christian.Schurig@esa.int

new satellite systems. Furthermore, outsourcing the maintenance of the ground stations to a *Ground Station as a Service* will allow satellite operators to focus on the satellites themselves. While there are already sophisticated scheduling algorithms for prioritizing important data in the presence of limited bandwidth (as described in the section on related work), they miss the element of a bidding and pricing scheme for competing satellite systems needed for such a service.

In this paper, we present an auction-based approach for coordinating and prioritizing download activities of a satellite constellation fit for a *Ground Station as a Service* system. In it, satellite operators bid on communication slots of ground stations and are billed correspondingly. If the occupancy is high, the prices automatically rise such that important, i.e., high-valued data is prioritized. We also demonstrate the power of our approach on benchmark constellations with more than 1000 satellites and a time window of 36 days: Our auction-based approach manages to retrieve ∼74% of the total data valuation, compared to merely ∼28% that are achieved by a standard greedy-type approach that is typically used by human operators with automation assistance.

II. RELATED WORK

A. Scheduling in General

Scheduling is a common problem in computer science with a vast amount of research and literature. A thorough introduction to the topic is given by Pinedo [2]. In addition to scheduling theory in general, there has been a considerable body of work on scheduling in the context of satellites. In her PhD thesis, Spangelo [3] considers operational challenges specific to small satellites in LEO, such as restricted onboard energy and data storage capacity.

Unfortunately, finding an optimal solution for even very restricted offline versions of the scheduling problem is known to be NP-hard. Therefore, exact solutions do not scale to large constellation sizes. However, for practical purposes, an algorithm that always achieves an optimal schedule is often not needed, motivating approximations algorithms and heuristics that are considered in the literature.

Scheduling problems that arise in practice are often inherently online in nature, i.e., scheduling decisions must be made without complete information. This also holds true for satellite operations: not all customer requests are known ahead of time, so it may be desirable to change the schedule after observing an event of interest. Furthermore, anomalies may occur during operation. Li et al. [4] consider an online scheduling variant with stochastic arrivals of urgent tasks and sequence-dependent setup times. They make a distinction between normal and urgent tasks. Rescheduling during orbit (between ground contacts) requires autonomous decision-making with limited/local information by the satellite. This is especially true if rescheduling is triggered because the satellite itself observed/measured an event of interest. Urgent customer request may be relayed via inter-satellite communication links (if available) from ground station to satellite. Stottler [5] reports a prototype implementation for a scheduler that incorporates case-based reasoning to automatically resolve conflicts. This conflict resolution is based on past decision-making and approval of human schedulers. A method for scheduling services in large satellite constellations is described by Marinelli et al. [6], in which types of scheduling problems are identified, and a time-indexed integer programming formulation is used. They also describe a practical heuristic approach based on Lagrangian relaxation and a Fix-and-Relax algorithm. The method was experimentally evaluated with promising results on the GALILEO project in research between Telespazio and the European Space Agency. Augenstein et al. [7] describe the scheduling algorithm that is used for the Terra Bella constellation. They consider the following problem setting and constraints: agile satellites that can change orientation, non-zero setup times for ground stations between different satellite contacts, a required minimum contact frequency for each satellite to transmit health/telemetry data and the ability for human operators to manually "lock-in" or "lock-out" a given contact. The objective is to balance image collection and data downlinking time, while maximizing the image collection of priority-weighted targets. Lee et al. [8] show a genetic algorithm for scheduling, which is designed with regard to the Korea Multi-Purpose Satellite (KOMPSAT) series.

B. Decentralized Auction Based Scheduling

As an alternative to centralized scheduling, Wellman [9] proposed *auction-based decentralized scheduling*, with competing agents bidding for resources. The schedule and the prices for the resources are determined by auction or market mechanisms. Prices may be purely virtual, but can also be used to determine actual usage fees. Agents work for their own advantage and may hold information private regarding their strategy. They have to evaluate the trade-offs of acquiring resources to maximize their objectives potentially with a limited budget.

In our case, satellites bid for communication slots of ground stations trying to maximize their downlinked data value while minimizing the overall costs. Therefore, the ground stations act as auctioneers. As in other auction-based mechanisms, the auctioneer continuously posts the price quotes to the agents. The agents communicate their bids iteratively, so that they can react to competing parties. After a fixed time window, the auctioneer computes the final schedule and price.

An auction may be differentiated across many parameters such as price determination algorithm, event timing, bid restriction, and intermediate price revelation. One of the most important distinctions is whether an individual auction allocates a single resource or several resources at once (*combinatorial auctions*).

The term *combinatorial auction* was popularized by Rasseneti et al. [10] in 1982. Later work on the complexity of the problem [11] showed that it is NP-hard to determine an allocation once all bids have been submitted to the auctioneer. These results lead to further research on the topic [12]. Solving distributed scheduling problems with market mechanisms has be proposed multiple times [13], [14]. Wellman [15] named his approach *market-oriented programming* (MOP).

Prior work has successfully applied market-inspired mechanisms to scheduling [16], [17] and other distributed resource allocation problems. Several have adopted the framework of general equilibrium theory and have found that the computational markets behave predictably when the conditions of the theory are met [15]. Additionally, the MOP approach was applied to a variety of discrete optimization problems. It has been successfully adopted to scheduling problems in manufacturing [18], [19], [20] and train or airport slot allocation [21], [22], [23]. Moreover, the benefits of the approach have been widely used in transportation services [12], [24].

Depending on the application of the scheduling problem, one can investigate bidding strategies that produce better solutions [25]. Bidding strategies like *iBundle* [26] (in which the price update is based on bid prices from unsuccessful agents) have brought great success to train scheduling problems [21].

III. PRELIMINARIES

Given the difficult conditions (with enormous distances and large amounts of data), ground stations and satellites use directed antennas for high-bandwidth communication. This requires satellites not only to be in range, but also necessitates precise adjustments of both sides, which can take up to two minutes. This also implies that satellites and ground stations have to be informed in advance of scheduled contacts. Because satellites in LEO are moving very fast (they can orbit earth in 90 minutes), the feasible contact windows are only a few minutes long. In addition, many satellite constellations operate in sun-synchronous orbits. As a consequence, satellite distribution is heterogeneous, with particularly high density near the poles, as polar orbits and ground stations increase the frequency of contact windows.

This leads to a basic model with the following underlying assumptions; some of them constitute slight but legitimate simplifications that are used in the following experiments.

- Ground station and satellites can establish a connection if the line of sight is not interrupted and a pass can be reliably predicted. Hence, for every satellite and ground station there is a continuously extended list of feasible contact windows (each having a minimal start and a maximal end time) in which data could be transmitted if there are no conflicts (see next point).
- Satellites and ground stations each need 2 minutes of adjustment time before each scheduled contact. During adjustment and contact, no other contacts can be performed.
- For simplicity, we use a homogeneous bandwidth. The amount of downloaded data hence can be measured in contact time.
- The onboard storage of each satellite is limited to the amount that could be downloaded in 10 000 seconds. If the memory is full, the least valuable data is automatically deleted. Data is created randomly (uniformly distributed size, value, and time).
- A contact can be performed if it is scheduled at least three hours in advance.
- The satellite operator has a knowledge of the data value on board of the satellites.

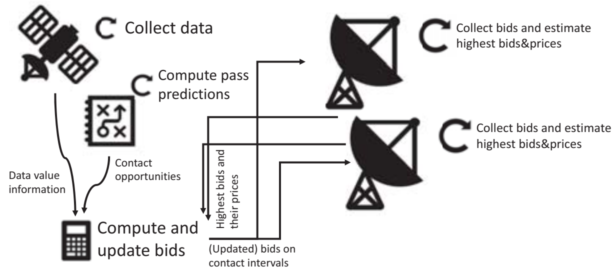

Fig. 1. Based on data value estimations and pass predictions, bids on contact intervals for each satellite are computed and sent to the corresponding ground stations. The ground stations constantly collect these bids and determine the highest bids and their current prices. Based on this information, the satellites can update their unsuccessful bids.

The last two assumptions are non-trivial but can be implemented by an additional low-bandwidth connection which are less restrictive than high-bandwidth connections and sufficient since the corresponding data is much smaller than the actual payload. But even without such an additional connection, data value can to some degree be predicted and supported by piggybacking informations during other contacts. Some satellites also do not need to be informed about contacts in advance because they automatically anticipate data from ground stations in reach and adjust correspondingly to some ground station.

To evaluate the scheduling quality, we consider in our experiments the objective of data and value rate as well as minimizing contact pauses. The data rate is the percentage downloaded of all generated data. The value rate is analogous but considers the value of the data.

IV. CONCEPT

In our algorithm (see Fig. 1), each satellite acts as a bidder who wants to buy contact times on ground stations to maximize the value of its downlinked data minus the price paid for the contacts. Ground stations act as auctioneers that accept bids on usage intervals and continuously determine the currently highest bids and their prices. A bid wins the auction when it remains the highest bid until a specified time before the beginning of its interval, such that there is enough time to actually schedule the contact. The currently highest bids are broadcast frequently, so satellites can move their bids into intervals with lower prices. All bids that are not marked as highest bid can be retracted or updated. The prices have to be fair and based on the actual demand (i.e., competition) such that they can be used as an actual fee in a *Ground Station as a Service* system. Note that neither the satellite nor the ground station will actually do any computations; these are carried out by separate computers, as the satellites do not have the required computational and communicational capabilities. Hence, the data value estimation for the satellites that is important to determine the bid amount is only based on meta-information and predictions.

There are two primary components:

1) An auctioneer's algorithm that determines the highest bids and their prices. It needs to be fast for continuous evaluation and transparent to proof its fairness.
2) A bidding strategy for satellites that can be different for each satellite.

While centralized scheduling algorithms like the one of Lee et al. [8] can be used to obtain a comparable (possibly even better) schedule, they miss a pricing scheme and do not allow different dynamic bidding strategies for competing satellite operators. Our approach provides not only a good scheduling quality, but also a fair pricing scheme and individual control for each satellite operator.

V. AUCTIONEER'S ALGORITHM

A. Optimal Bid Selection

If a ground station is given a set of offers for intervals, it is not obvious how to select the highest bidders. While for independent resources one could simply always take the highest offer, this is not the case if the resources are overlapping as it is the case for time intervals: Multiple short intervals with low offers can together outweigh a high offer for a long interval. While one could still accept the interval with the highest offer or the highest relative offer (price/length), either strategy would encourage to either buy very long intervals or to drive out the competition by very short intervals. The most reasonable approach is to select the intervals as winning bids that in combination are willing to pay the most. The optimal selection of winning bids can be computed efficiently, which allows us to repeat this procedure at a high frequency.

Theorem 1: Given a set of n intervals and corresponding prices. We can compute the optimal selection of intervals such that no two selected intervals are intersecting and the sum of prices is maximal in $O(n \log n)$ time or $O(n)$ if the intervals are already sorted by their end.

Proof: This problem is also known as the *Weighted Interval Scheduling Problem*. It can be solved in $O(n \log n)$ resp. $O(n)$ via a simple dynamic program [27, Chapter 6.1]. ■

Let $\texttt{WIS}(B, p)$ denote the optimal Weighted Interval Schedule for intervals B and prices/weights p. If there are multiple such solutions, return the one with the most intervals (can be easily integrated into the dynamic program).

B. Price estimation

A natural approach to pricing a winning bid is the Vickrey principle, with the winning bid paying the price of the second highest bid. Applying this to our case with overlapping intervals requires some care, as there is no such thing as a clear "second highest bid"; e.g., the competition may be based on a long interval that has been outbid by multiple smaller intervals. To get a fair pricing, we set every price initially to some small value (such as zero or one) and increase the price only if the interval is not among the winners and their offer is not yet reached. This process ends when none of the prices of the losing bids can be increased anymore because they reached their current (maximum) offer. If the price of a winning bid is not limited by its offer, it would also have the same price even if its offer had been higher. This achieves the Vickrey principle, with the price mainly determined by the competition and not by the own offer. The algorithm is described in Alg. 1, an example is given in Fig. 2.

Data: A set B of bids with begin, end, and offer. A map $\texttt{price} : B \to \mathbb{R}_0^+$ with (initial) prices. A price increase function $\texttt{inc} : \mathbb{R}_0^+ \to \mathbb{R}_0^+$
Result: A subset $W \subseteq B$ of winning bids and map of prices $\texttt{price} : W \to \mathbb{R}_0^+$

```
W = WIS(B, price);
while ∃b ∈ B \ W : price[b] < b.offer do
    for b ∈ B \ W do
        price[b] = min(inc(price[b]), b.offer);
    end
    W = WIS(B, price);
end
return W, price;
```

Algorithm 1: Auctioneer's Algorithm.

There are multiple options to increase the price, e.g., adding a constant value, a percentage of the current price, or a percentage of the maximum offer. Each of these solutions results in different runtimes.

Theorem 2: Algorithm 1 with n bids and the highest offer being h terminates after at most

- $O(n^2 * h)$ steps for constant increments, e.g. $+1$.
- $O(n^2 * \log h)$ steps for proportional increments, e.g. $+20\%$ (initial price has to be positive).
- $O(n^2)$ steps for e.g., $+0.05 * b$.offer.

Proof: In every step, at least one price is increased and every bid can only be increased $O(h)$/$O(\log h)$/$O(1)$ times. Sorting the bids for the end takes $O(n \log n)$ and reevaluating W takes $O(n)$. ■

Lemma 1: For a set of bids B, Algorithm 1 returns the same winning bids as $\texttt{WIS}(B, b \to b.\text{offer})$ if the solution is unique.

Proof: Further increments of the prices in Algorithm 1 do not change the winning bids, because all losing bids cannot increase their offer to become more attractive. The solution is thus identical to $\texttt{WIS}(B, b \to b.\text{offer})$. ■

In practice one can achieve a significant improvement by estimating the winning and losing bids first (because they do not have to pay anyways). Then we directly set the losing bids to their maximum price and prevent many alternating increments.

Theorem 3: If we set the price of the losing bids directly to their maximum and assuming the bids are sorted in

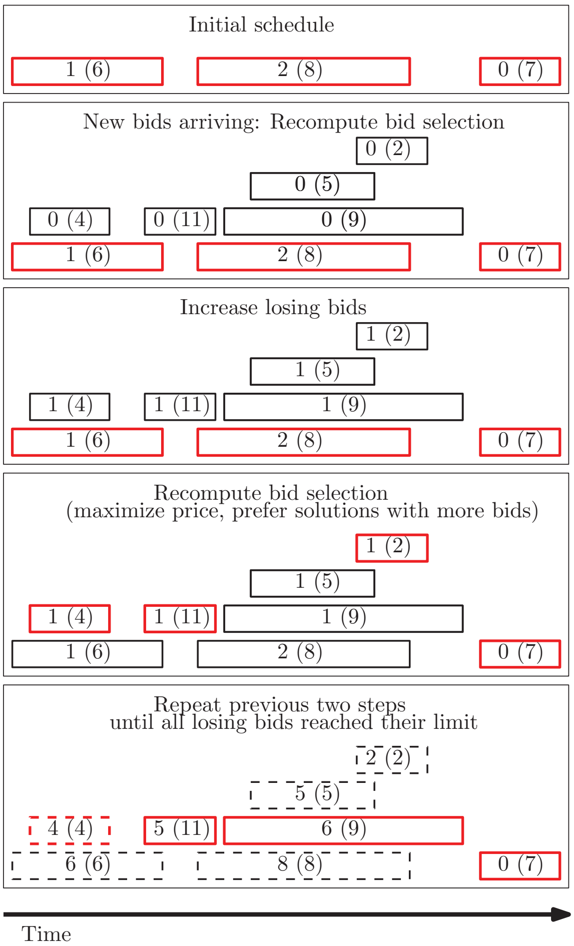

Fig. 2. Example of the auctioneer's algorithm. Every block is a bid for a time interval. It contains the current price and in brackets the limit. Red bids are the current bid selection. Dashed bids have reached their limit.

advance, Algorithm 1 with n bids and the highest offer being h terminates after at most

- $O(n * h)$ steps for constant increments, e.g. $+1$.
- $O(n * \log h)$ steps for proportional increments, e.g. $+20\%$ (initial price has to be positive).
- $O(n)$ steps for e.g., $+0.05 * b$.offer,

Proof: Let W' be the winning bids, determined by $\texttt{WIS}(B, b \rightarrow b.\text{offer})$. In every step, $W \cap W'$ never decreases. Let w' be the last bid to be added to $W \cap W'$. It is incremented in every round, because it is losing in every round except the last one. This is only possible $O(n)/O(\log n)/O(1)$ times. ■

By using a binary search to find the increment on which the solution changes can improve the runtime further. However, in both cases the prices can drastically differ to the original version.

VI. BIDDING STRATEGY

Every satellite can have its own strategy that best fits its purpose. A simple strategy can be to consider all feasible intervals on all ground stations and estimate their gain by using the value of the currently available (and not already scheduled) data and the prices of the intersecting highest bids. Then one bids on the interval with the highest expected gain and marks this interval and the corresponding data as scheduled (if this bid loses, this is undone). The maximum offer should not be the estimated value but only some percentage above the currently estimated price because at some price, another interval has a higher gain. This is repeated until there are no more intervals with positive gain on any ground station. The relative value of later bids decreases as the "good data" is already used for the previous bids. In order to discretize the set of intervals, we only consider those that begin at event points, such as the beginning of other highest bids, as well as periodical event points (e.g., every 5 minutes). This basic approach can be further refined by a number of methods, as follows.

- Use a prediction for future data value gain until the beginning of the considered interval.
- Consider the success probabilities of the previous bids for your value estimations and possibly even bid for conflicting intervals.
- Use the "second best opportunity" to estimate the maximum one would pay for an interval before switching to the second best. As intervals have different lengths, this is not trivial.
- Considering that intervals farther in the future may be subject to higher price increases.
- Multiple lower bids can outbid a higher bid. However, these bids are usually from different (not cooperating) satellites, so it is reasonable to occasionally try bids that do not seem to have a chance.

Constellation	1	2	3	4
Inspiration	Galileo	Artificial	Dove	OneWeb
Number of S/C	39	40	400	1080
Orbit region	MEO	LEO	LEO	LEO
Inclination/deg	56	85	97	87.9
Eccentricity/-	0.001	0.001	0.001	0.001
Number of ground stations	2	6	5	6

TABLE I
SATELLITE CONSTELLATIONS USED FOR EXPERIMENTAL EVALUATION.

VII. EXPERIMENTAL EVALUATION OF SCHEDULE QUALITY

To validate the schedule quality of our method, we have evaluated four different satellite constellations, as shown in Table I. They contain the Galileo constellation, a slightly enlarged constellation version of the Dove constellation of Planet, and one modeled after the planned OneWeb constellation. The Galileo constellation is in medium earth orbit and is significantly slower, i.e., passes and pauses are much longer than for the LEO constellations. Each of the four cases was simulated for a schedule of 36 days to compute realistic contact windows. The satellites continuously generate data whose size and value is determined by a uniform probability distribution.

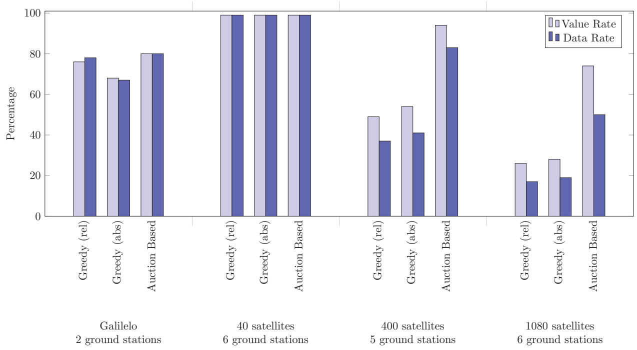

Fig. 3. Data and value rate for different constellations over 36 days. For the first two experiments with few satellites, the performance of the greedy approaches and our auction-based approach are relatively similar. However, the performance of the greedy approaches drops significantly for the two larger experiments while our auction-based approach still achieves relatively good rates.

A. Greedy Algorithm

To get a comparison for the schedule quality, we also implemented a greedy algorithm. It schedules for a time interval (e.g., 1-3 hours) all intervals by iteratively selecting the contact with the highest (relative) download value until no more contacts can be scheduled. We use one variant with the highest absolute value and one variant with the highest relative value (value/length). On tie, the shorter contact is chosen such that conflicts are minimized. This is fairly similar to how a human approaches this problem.

B. Experimental Results

In Fig. 3 one can see that for the second constellation, almost all data can be downloaded, even with a greedy approach. For the third and fourth constellation, our auction-based approach still achieves a relatively high rate, while the greedy approach drops considerably. The reason is that it is no longer possible to download all data, so an algorithm has to make the best choices; this also implies an increasing difference between value rate and data rate. An excerpt of a schedule is shown in Fig. 5. The results for Galileo can be explained by the long contact-less intervals that keep a lot of the data out of reach.

C. Contact pauses

Besides the value of the data, one is also interested in minimizing the maximal pauses between two contacts, i.e., the continuous time without contact. Both greedy and our auction-based algorithm can have relatively long pauses without further adjustments. However, the bidder strategy of the satellites can easily be adapted to keep these pauses small. If the last scheduled contact of a satellite is too old, the estimated value is increased based on the time distance such that it continuously increases. The satellite will now give continuously increasing bids, until it wins a bid after which this "boost" is reset. The result are much shorter maximum contact pauses for the satellites (see Fig. 4), while the downloaded value is only slightly worse. This shows the flexibility of our approach.

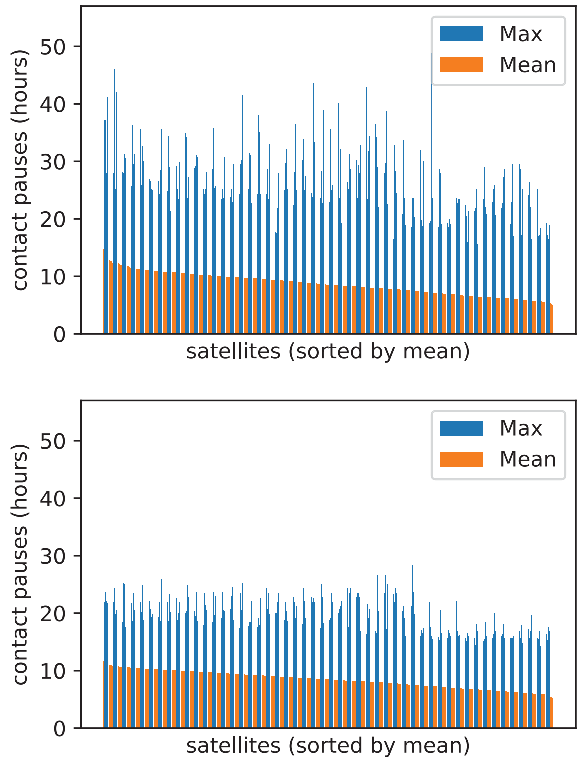

Fig. 4. If one wants to keep the contact pauses of a satellite short, one can increase the bids with the length of the current contact pause in order to easily minimize those. (Top) Contact pauses without optimization for the Dove-like constellation. (Bottom) Contact pauses with optimization. The maximum pause has almost been cut in half while barely reducing the data and value rate.

VIII. Conclusion

We have provided an auction-based method for optimizing the download schedules for large-scale constellations of spacecraft. Our simulation results demonstrate the power of our approach. Future developments can be expected from inter-satellite communications, which will give rise to even more complex perspectives of collecting and aggregating data, as well as more flexibility than strategies involving fixed ground stations. We are optimistic that similar methods will be useful and increase in importance as the challenges for automated spacecraft continue to increase.

References

[1] D. L. Bindschadler, C. A. Boyles, C. Carrion, and C. L. Delp, "MOS 2.0: The next generation in mission operations systems," in *SpaceOps Conference*. NASA, 2010.

[2] M. L. Pinedo, *Scheduling: theory, algorithms, and systems*. Springer, 2016.

[3] S. C. Spangelo, "Modeling and optimizing space networks for improved communication capacity." 2013.

[4] G. Li, L. Xing, and Y. Chen, "A hybrid online scheduling mechanism with revision and progressive techniques for autonomous earth observation satellite," *Acta Astronautica*, vol. 140, pp. 308–321, 2017.

[5] D. Stottler, "Satellite communication scheduling, optimization, and deconfliction using artificial intelligence techniques," in *AIAA Infotech@ Aerospace 2010*, 2010, p. 3424.

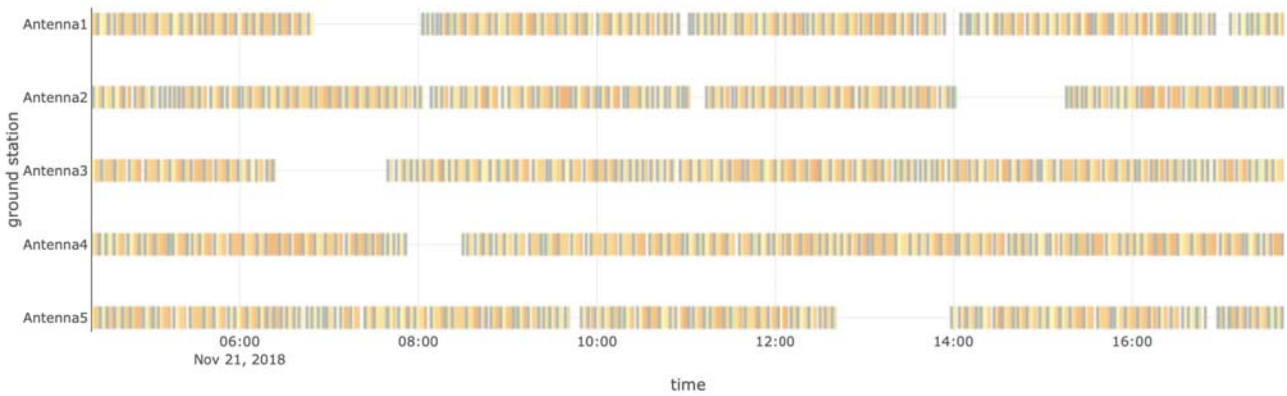

Fig. 5. An excerpt of the schedule for the Dove-like instance. The grey strips are adjustments, the colored strips visualize the value of downloaded data packages. Note that even for 400 satellites, there are still pauses in which no satellite is in range of the ground station. During the other times, the schedule is packed tightly. The larger OneWeb-like instance does not have any pauses.

[6] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, "A lagrangian heuristic for satellite range scheduling with resource constraints," *Computers & Operations Research*, vol. 38, no. 11, pp. 1572–1583, 2011.

[7] S. Augenstein, A. Estanislao, E. Guere, and S. Blaes, "Optimal scheduling of a constellation of earth-imaging satellites, for maximal data throughput and efficient human management." in *ICAPS*, 2016, pp. 345–352.

[8] J. Lee, H. Kim, H. Chung, and K. Ko, "Genetic algorithm–based scheduling for ground support of multiple satellites and antennae considering operation modes," *International Journal of Aeronautical and Space Sciences*, vol. 17, no. 1, pp. 89–100, 2016.

[9] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason, "Auction Protocols for Decentralized Scheduling," *Games and Economic Behavior*, vol. 35, no. 1-2, pp. 271–303, 2001.

[10] S. J. Rassenti, V. L. Smith, and R. L. Bulfin, "A combinatorial auction mechanism for airport time slot allocation," *The Bell Journal of Economics*, pp. 402–417, 1982.

[11] M. H. Rothkopf, A. Pekeč, and R. M. Harstad, "Computationally Manageable Combinational Auctions," *Management Science*, vol. 44, no. 8, pp. 1131–1147, 1998. [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/mnsc.44.8.1131

[12] Y. Sheffi, "Combinatorial auctions in the procurement of transportation services," *Interfaces*, vol. 34, no. 4, pp. 245–252, aug 2004.

[13] S. Pal, S. R. Kundu, M. Chatterjee, and S. K. Das, "Combinatorial reverse auction based scheduling in multirate wireless systems," *IEEE Transactions on Computers*, vol. 56, no. 10, pp. 1329–1341, oct 2007.

[14] H. Harb, J. N. Paprott, P. Matthes, T. Schütz, R. Streblow, and D. Müller, "Decentralized scheduling strategy of heating systems for balancing the residual load," *Building and Environment*, vol. 86, pp. 132–140, 2015.

[15] M. Wellman, "A market-oriented programming environment and its application to distributed multicommodity flow problems," *Journal of*

[20] Q. lin Guo and M. Zhang, "An agent-oriented approach to resolve scheduling optimization in intelligent manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 1, pp. 39–45, 2010.

Artificial Intelligence Research, vol. 1, pp. 1–23, 1993.

[16] G. Confessore, S. Giordani, and S. Rismondo, "A market-based multi-agent system model for decentralized multi-project scheduling," *Annals of Operations Research*, vol. 150, no. 1, pp. 115–135, 2007.

[17] W. Walsh, M. Wellman, P. Wurman, and J. MacKie-Mason, "Some economics of market-based distributed scheduling," *Proceedings. 18th International Conference on Distributed Computing Systems (Cat. No.98CB36183)*, pp. 1–14, 1998.

[18] V. Kumar, S. Kumar, M. K. Tiwari, and F. T. Chan, "Auction-based approach to resolve the scheduling problem in the steel making process," *International Journal of Production Research*, vol. 44, no. 8, pp. 1503–1522, 2006.

[19] S. Giordani, M. Lujak, and F. Martinelli, "A distributed multi-agent production planning and scheduling framework for mobile robots," *Computers and Industrial Engineering*, vol. 64, no. 1, pp. 19–30, 2013.

[21] D. C. Parkes and L. H. Ungar, "An auction-based method for decentralized train scheduling," in *Proceedings of the fifth international conference on Autonomous agents - AGENTS '01*, 2001, pp. 43–50.

[22] R. Borndörfer, M. Grötschel, S. Lukac, K. Mitusch, T. Schlechte, S. Schultz, and A. Tanner, "An auctioning approach to railway slot allocation," *Competition and Regulation in Network Industries*, vol. 1, no. 2, pp. 163–197, 2006.

[23] S. J. Rassenti, V. L. Smith, and R. L. Bulfin, "A combinatorial auction mechanism for airport time slot allocation," in *Handbook of Spectrum Auction Design*, 2017, pp. 373–390.

[24] P. Davidsson, L. Henesey, L. Ramstedt, J. Törnquist, and F. Wernstedt, "An analysis of agent-based approaches to transport logistics," pp. 255–271, 2005.

[25] D. M. Reeves, M. P. Wellman, J. K. MacKie-Mason, and A. Osepayshvili, "Exploring bidding strategies for market-based scheduling," in *Decision Support Systems*, vol. 39, no. 1, 2005, pp. 67–85.

[26] D. C. Parkes, "iBundle," in *Proceedings of the 1st ACM conference on Electronic commerce - EC '99*, 1999, pp. 148–157.

[27] J. Kleinberg and E. Tardos, *Algorithm Design*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.