A Scheduling System with Redundant Scheduling Capabilities

Marco Schmidt and **Klaus Schilling**
University of Wuerzburg
Wuerzburg (Germany)
schmidt.marco@informatik.uni-wuerzburg.de
schi@informatik.uni-wuerzburg.de

Abstract

The concept of ground station networks evolved in the recent years in many different fields as a reasonable tool for satellite applications. Especially in the field of academic ground station networks, grown from small satellite projects at universities all over the world, new innovative ideas were developed to enhance the operation of small low earth orbit satellites. One important aspect, which has not been considered so far, is an essential difference between academic networks and classical ground station networks with respect to scheduling requirements . These differences are mainly conducted by the architecture and non-commercial character of academic networks, resulting in different scheduling requirements like flexibility and redundant scheduling. This work defines a mathematical description of the scheduling problem appearing in academic ground station networks, referred as the Redundant Request Satellite Scheduling (RRSS) problem. Furthermore important differences compared to the classical problem formulation are elaborated. Additionally the paper introduces a new scheduling approach tailored for the special needs of this problem.

Introduction

Ground station networks offer new interesting capabilities for satellite operation or distributed space applications. In the last years several projects with the objective to establish ground station networks on a world-wide basis were started (Shirville and Klofas 2007)(Nakamura and Nakasuka 2006). Many of these projects were initiated from the small satellite community, including many academic institutes with their own ground station to track satellites. The objective of these ground station networks is to extend the contact time with the corresponding satellites and to increase the utilization of the ground stations through sharing resources (i.e. ground stations). An important issue for sharing resources in ground station networks is scheduling, i.e. to find an optimal (or at least suitable) assignment of satellites to ground stations to increase the utilization. The problem of finding an optimal solution of contact window assignment is referred as Satellite Range Scheduling (SRS) Problem (Schalck 1993) (Barbulescu et al. 2002). The problem instances of SRS are very often oversubscribed (Barbulescu, Kramer, and Smith 2007), an optimal allocation of ground stations to satellites is not possible, therefore unsatisfied communication requests are unavoidable. Especially in the field of academic ground station networks the huge number of small satellites can increase the problem to strongly oversubscribed level. State of the art scheduling techniques for SRS have been investigated extensively, nevertheless these approaches are very often not appropriate for the requirements of academic ground station networks. One major issue is the non-commercial character of academic ground station networks, which results in different scheduling needs: The scheduling requests in academic satellite projects are much more flexible and dynamic. Furthermore the scheduling objectives differ in many ways (c.f. section 2), as the participants of these networks are sharing their resource without commercial interest. Standard scheduling algorithms are very often not applicable for these kind of problems, therefore a scheduling system which fulfills the special needs of an academic ground station network was developed (Schmidt, Rybysc, and Schilling 2008). The mathematical description of that special problem occurring in academic ground station networks is defined, as well as an approach to solve the problem is presented in this work.
The paper is arranged as follows. Section 2 starts with some basic requirements of redundant request scheduling, which has not been considered in classical problem formulations so far. The mathematical problem description of the underlying scheduling problem, introduced here as the Redundant Request Satellite Scheduling (RRSS) problem, is described in more detail in section 3 , an approach to solve the RRSS problem is handled in section 4 . A conclusion and and some remarks about future work are presented in the last section.

Flexible and redundant request scheduling

Previous work in this field was very often related to the ground station networks of huge space agencies (Damiani et al. 2007) (Barbulescu et al. 2004), dealing with a huge amount of satellite requests every day. The main objectives of these networks is to increase the utilization of the ground station networks or to satisfy as much requests of costumers as possible. The work described in this paper has its origin in an academic ground station network, initiated by the CubeSat community. Main interests of this network is to provide access to the satellites for the corresponding satellite developers. This access is granted without commercial interest.

Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Therefore we critically investigated the requirements for a academic ground station scheduling system, with the result that a basic assumption is totally different, when comparing classical ground station networks with academic ground station networks. These differences are: A request from a user in a classical ground station network wants a communication window with his satellite exactly like specified in the request. This communication window is normally used for an experiment or to transmit telemetry. The user normally requests a fixed time interval for communication from the provider of the ground station network. Assignment of this communication window satisfies the needs of the user. In an academic ground station network the requests from the users are much more flexible and not strictly defined. The communication of an user is very often not requested as a fixed interval for a predefined experiment, the user rather wants to have at least one possibility to communicate with his satellite, but if there is the possibility to have more contact windows, this solution is preferred. Especially in the Launch and Early Operations Phase (LEOP) the satellite developers request as much time as possible to retrieve telemetry from the satellite (Schmidt and Schilling 2008). This can be observed quite often when a batch of several pico satellites is launched together with the same launch vehicle: Directly after launch all the satellite developers request as much contact time with their satellite as available, but they would need at least one contact window to check if everything runs normal on the satellite. Furthermore the users are not interested which ground station tracks the satellite at what time (as these ground stations are very similar in architecture and in general interoperable), so it is possible to shift a communication to another ground station or use a different contact window for communication. Of course the resources (ground stations) are not always exchangeable, but they are at least more flexible usable than in classical ground station networks. For example does a satellite developer do not care if he communicates with its satellite through a ground station in Wuerzburg, Aalbourg or Tokyo, as an additional contact window on a foreign ground station is already better than no contact. In a classical ground station network the situation is different: A customer would pay for exactly one contact window with its satellite over a ground station which supports the communication link, it is in general not possible to change to any other ground stations of the network, as they are not interoperable. A second contact window will be only assigned if the costumer requests a second contact window and also will pay for it. To summarize the main differences between classical and academic ground station networks: Academic ground stations networks are more flexible, the contact windows can be shifted to other participating ground stations, time limits are not that strict as the participants are not paying for assigned contact windows. Furthermore the requested communication time is not restricted to one communication window, it is possible (or even desired) to include more contact windows for redundancy purposes and to increase contact time. Both differences in scheduling are triggered by the circumstance that no financial interest stands behind the academic ground station networks.

The aspect of redundant scheduling of contact windows has not been considered so far. It is problematic for a scheduling system if the exact number of desired contact windows is not known from before. Many standard scheduling methods for the SRS problem try to satisfy as many requests as possible, which is hard to do if a request is not automatically satisfied if exactly one contact window was assigned. Therefore a problem description and an approach to solve the redundant scheduling problem was developed.

Mathematical problem description

The problem of ground station scheduling can be described in few words: To find an optimal assignment of ground stations to a number of satellite requests, submitted by different users. This very vague description will be more formalized in this section. The entities of the scheduling problem are ground stations, users and satellites, which are related to each other over communication requests. The most important entity and central to this problem description is the definition of a communication request R (further only referred as request), it describes the request of an user U for a contact time with a satellite S. The ground station on which this contact takes place is not defined from the user before, the ground station network itself guarantees seamless data flow through the connected ground stations (for example over Internet). Therefore a request R_i is clearly defined through

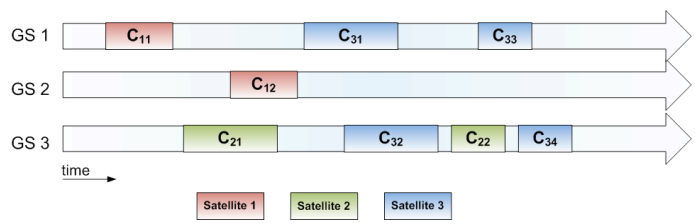

Figure 1: Calculated contact windows Cij

$$R_i = \{S, U, t_s, t_e, dur, Rd\} \tag{1}$$

The parameters of a request R_i are the requested satellite S, which has a defined orbit, an user U, who submitted the request, an earliest start time t_s and a latest end time t_e, which describe the time frame desired from the user (for example in the next two days). The duration dur describes the length of the requested contact window (in minutes) and the parameter Rd describes the maximum degree of redundancy. As explained in the last section , the aspect of including redundant contact windows is essential for this problem formulation. It is possible to define an upper bound for the number of redundant scheduled contact windows. If no upper bound for the maximum redundancy is defined, it is possible to assign as many contact windows as available to a given request. We define the possible values for Rd as

$$Rd = \begin{cases} n \in N_+ \text{ maximum of n contact windows} \\ -1 \text{ if no upper bound is set} \end{cases} \tag{2}$$

In the problem description all basic entities, i.e. users, ground stations, satellites and requests, are associated with priorities. The introduction of priorities was decided to be able to express the contribution of an user or satellite to the

ground station network. In the further text the priorities P are marked with a subscript for the corresponding component, so P_S stands for the priority value of the satellite, P_R for the request and P_U for the user respectively. With the defined structure of requests, satellites and users the available contact windows C_{ij} can be calculated. Each request R_i has a certain number of contact windows associated, which is determined through the orbit geometry and the start and end times (t_s and t_e) respectively. These j contact windows of a request R_i are distributed over several ground stations. A contact window C_{ij} is defined through

$$C_{ij} = \{R_i, t_{AOS}, t_{LOS}, G\} \tag{3}$$

Each contact window has an associated request R_i, the contact window could be used to accommodate a time interval of dur minutes, and an associated ground station, which this contact window is valid for. The values t_{AOS} and t_{LOS} describe the time interval where contact between the satellite S and ground station G is possible. It should be emphasized that t_{AOS} and t_{LOS} are not the same as the start and end times t_s and t_e of the request: A request could ask for a 10 minute contact time in the time from now (t_s) until next week (t_e), an appropriate contact window C_{ij} could start tomorrow morning (t_{AOS}) and end 15 minutes later ($_{LOS}$). Figure 1 illustrates a scenario with 3 satellites, the contact windows are distributed over 3 ground stations. The first index describes the associated request (e.g. all blue contact windows belong to request number 3), the second index numbers consecutively the contact windows belonging to the same request. The calculation of these contact windows can be performed automatically with an orbit prediction software (for example *STK* or *predict*). After calculation of the available contact windows C_{ij}, the problem is to find an optimal assignment of these contact windows to the requests R_i, which is problematic due to overlapping contact windows. This problem is not trivial and an algorithm to find an optimal solution in reasonable time is not known, due to the NP-complete characteristic of that problem (Barbulescu et al. 2004). Furthermore this problem is in many cases oversubscribed, many contact windows are overlapping with each other and it is not possible to decide which contact window should be excluded from the actual schedule or shifted to another ground station (which could collide with another contact window again).
An important issue, which should be mentioned again, is the assignment policy of the contact windows C_{ij}. In classical ground station networks, a request is satisfied as soon as one contact window out of the j available contact windows of C_{ij} was included in the final schedule. In the RRSS problem domain the aim is to include more than 1 contact windows (if possible) in the schedule for redundancy purposes.

The scheduling system

The RRSS problem formulation, described in the previous section, has not been considered so far, especially the aspect of scheduling redundant contact windows for a user request is not possible in classical problem formulations. In this section an approach is presented which handles the aspect of redundancy with a customized objective function. This objective function is necessary to search for a satisfying scheduling solution.

Scheduling objective function

The objective of the scheduling system is defined as a cost function γ (see equation 4), which has to be maximized. This cost function calculates a value for the given schedule σ, consisting of all calculated contact windows C_{ij} and their assignment. The cost function γ contains two terms γ_1 and γ_2, they are described in more detail in the following paragraphs.

$$\gamma(\sigma) = \gamma_1 - \gamma_2 \tag{4}$$

γ_1 is the weighted sum of the different assigned priorities. Maximizing this term means to include as many "high priority" contact windows C_{ij} as possible, it can be written as

$$\gamma_1 = \sum_{\forall C_{ij}} \left(\pi(C_{ij}) \cdot C^b_{ij}\right) \tag{5}$$

with C^b_{ij} defined as

$$C^b_{ij} = \begin{cases} 1, \text{ if } C_{ij} \text{was integrated into the schedule} \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

and

$$\pi(C_{ij}) = w_R \cdot P_R + w_G \cdot P_G + \ w_S \cdot P_S + w_U \cdot P_U \tag{7}$$

The function $\pi(C_{ij})$ calculates the total priority value out of weighted entity priorities (P_R, P_B, P_S, P_U). Thus, it is possible to control the contribution if the different scheduling entities in a very fine grain. For our work the weights (w_R,w_G,w_S,w_U) were chosen to a fixed value of 1.

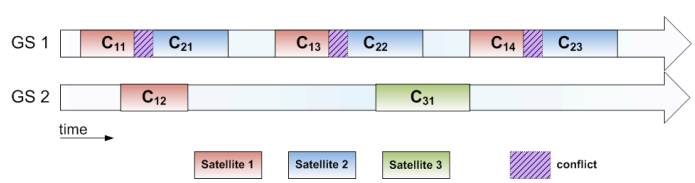

Figure 2: Two satellites in a similar orbit

Maximizing the γ_1 term means to include as many high priority contact windows as possible, not considering to include low priority requests. As the problem formulation states, it is possible that a request gets more than one contact window assigned, therefore it could be possible that one request receives two contact windows and another one no contact window at all, even if both request would be satisfiable. The problem here is that redundant scheduled contact windows of high priority requests will preempt contact windows of low priority requests. To avoid "unfair" distribution of contact windows, the γ_2 term is subtracted from the the γ_1 function. The aspect of fairness, here interpreted as an equal distribution of contact windows for requests, is integrated by the term γ_2, it can be seen as a penalty for unfair distribution

of contact windows. The aim of this term is to avoid situations, where two equal requests are treated unfair with respect to overlapping windows. The scenario depicted in figure 2 shows a situation where two satellites fly almost in the same orbit shortly after each other, which results in conflicting contact windows each revolution at ground station 1 (GS 1). If both satellites have the same priority, the γ_2 term guarantees that the associated requests receive the same amount of contact windows. The γ_2 term is defined as:

$$\gamma_2 = \sum_{1 \leq k \leq i} \left(\lambda^{\kappa(R_k)} \right) \tag{8}$$

where λ is a positive integer ≥ 1 and the function $\kappa(R)$ is defined as

$$\kappa(R_i) = R^b_{max} - R^b_i \tag{9}$$

$$R^b_i = \sum_j (C^b_{ij}) \tag{10}$$

The term R^b_{max} describes the maximum possible value over all R^b_i (amount of assigned contact windows for request i). From this definition it is clear that $\kappa(R_i)$ is ≥ 0, which is an important property for the function γ_2. The aim of the objective function γ_2 is to distribute redundant contact windows equally over the requests. It can be shown that γ_2 is minimal, if the contact windows of all requests R_i are equally distributed. As equation 8 is a sum, the whole term is minimal if the summands are pairwise minimal. It can be shown that two summands (λ^k) are minimal for the contact windows C_{ij} and C_{kj} of two requests R_i and R_k, if they are equally distributed. For the simple case that these two requests have no overlapping (conflicting) contacts, the κ term would be minimal in both cases (as R^i_{max} is an upper bound). Therefore γ_2 is also minimal for that case. For the schedule itself, this means that a maximization of the complete objective function γ has to minimize the term γ_2 and to maximize the term γ_1. For the case that both request R_i and R_k have overlapping contact windows, it can be shown that the same behavior holds if the redundancy between these two requests is equally distributed (Schmidt 2008). So far undiscussed is the parameter λ, which influences the "importance" of the γ_2 term on the overall objective function. If λ is chosen to a small integer, the objective to distribute the redundancy equally between requests has only small influence on the objective to include as many high priority contact windows (γ_1) as possible, so γ_2 is comparably smaller than γ_1. But if the λ value is set to a bigger integer, the objective to equally distribute redundancy overwhelms the γ_1 objective. Therefore a trade off between these two objectives has to be found, in our experiments we empirically determined a value for λ of 3 as appropriate.

Search algorithms

The definition of a scheduling cost function is a necessary prerequisite to search for satisfying solution on a set of schedules. In the next step two search strategies are introduced, which are used to find satisfying solutions to a given problem instance. This means to find a suitable assignment of the contact windows C_{ij} to the requested satellites and to maximize the overall cost function γ.

Branch-and-Bound algorithm

The Branch-and-Bound strategy is a general algorithm to find solutions in optimization problems. It is not a search algorithm itself, it is a technique which divides a problem in subproblems (branch) and tries to evaluate these problems with respect to an upper limit (bound). Solutions of subproblems which promise to have bad overall performance are used to limit the search space with the help of a decision tree. This strategy of limiting the search space is reasonable, as it is not possible to evaluate all possible schedules (the problem is NP-complete). The search algorithm used by the Branch-and-Bound strategy is the depth-first search. Problematic for the Branch-and-Bound algorithm is, if a huge amount of subproblems have to be solved, which results in large runtime and memory demands.

Hillclimber

The hillclimbing search is a simple, heuristic algorithm, which tries to search from a given starting point in the search space (i.e. a schedule) for a better solution. The algorithm searches as long as the objective function can be increased, as soon as it is not possible to further increase the objective function, the hillclimbing algorithm stops the search. Disadvantage of this search strategy is the problem of remaining in local maxima, to overcome this problem stochastic means have to be included. In our case the starting point in the search space is chosen randomly. The implemented version of our hillclimbing algorithm tries to optimize the cost function introduced in section . The performance of the hillclimber algorithm on other scheduling problems related to ground station networks depends a lot on the underlying problem. In (Barbulescu et al. 2007) the overall performance is poor compared to other search algorithms, in (Globus et al. 2004) the hillclimber outperforms other heuristics. In this case the hillclimber was mainly implemented to have an comparison to the Branch-and-Bound algorithm.

Scheduling a demonstration scenario

To demonstrate the implemented system a real world scenario was used. As this scheduling system originated from a small satellite project at the University of Wuerzburg, the experiment shows a typical scenario with several small satellites. The requests are listed in table 1, they are defined according equation 1. The start and end times (t_s and t_e) were set to a time interval of 5 hours. Of course it is possible to generate schedules for larger time frames, but in this experiment an oversubscribed problem instance with many conflicting contact windows should be shown, therefore the time limits were set only to small interval of 5 hours. Furthermore in this experiment all priorities were set to the same value of 10 (P_U, P_S, P_G, P_R) for better comparison of the results.

The participating ground stations of the network are located

Request	User	Satellite	dur	Rd
R0	U0	Cute-1	10 min	-1
R1	U1	Cubesat XI-IV	10 min	-1
R2	U2	Cubesat XI-V	10 min	-1
R3	U3	CP4	10 min	-1
R4	U4	Cape-1	10 min	-1
R5	U5	JAK	10 min	-1

Table 1: Requests of the demonstration scenario

Request	Available	Scheduled
R0	11	4
R1	11	4
R2	9	9
R3	7	6
R4	11	8
R5	2	1
Σ	51	32

Table 2: Assignment of contact windows

in Europe (Denmark, Sweden, Germany) and Asia (Japan). The location of three ground stations near to each other shows an overlapping scenario. The Two Line Elements (TLE) of 6 satellites are available and used to calculate the contact windows C_{ij}, an orbit prediction software automatically calculates the parameters t_{AOS} and t_{LOS} of equation 3. For this scenario, consisting of 6 requests related to 6 satellites and 4 different ground stations, 51 contact windows were calculated by the orbit prediction software. From these 51 available contact windows are 6 too short to accommodate a 10 minute contact window. Furthermore conflicts exist between the calculated C_{ij}, in this scenario 42 conflicts were determined. The proposed scheduling system found a suitable schedule, the result is shown in table 2.

The result from table 2 shows that all request received at least one contact window, so all requests are satisfied. Due to the conflicting contact windows in this scenario it was not possible to assign all 51 available contact windows, but redundant contact windows were assigned to all requests (except R5, which has only two available contact windows). Furthermore an equal distribution of redundant contact windows was achieved. Even if it seems "unfair" that request R2 received all available 9 contact windows while R0 and R1 received only 4 windows, this can be explained very simple. Due to the orbits of the satellites almost no conflicts were present for the request R2, furthermore too short contact windows for R0 and R1 were present to accommodate a contact of the required 10 minutes. The output of the scheduling system on its graphical user interface is depicted in figure 3 on the 5 hour interval from 8:00 until 13:00.

This scenario shows the capabilities of the proposed scheduling system. The redundant scheduling capability is very useful in scenarios appearing in educational ground station networks, were it is quite common that several satellites are in similar orbits and all of them want to have as much available contact time as possible. This is especially true during Launch and Early Orbit Phase (LEOP), when several satellites were brought into orbit with the same launch vehicle. Furthermore enables the short execution time of the scheduling application a fast rescheduling. Flexible scheduling is very valuable, especially in dynamic environments, like it is typical at universities. The system can therefore improve the performance as it satisfies the special needs of academic ground station networks better than classic scheduling systems.

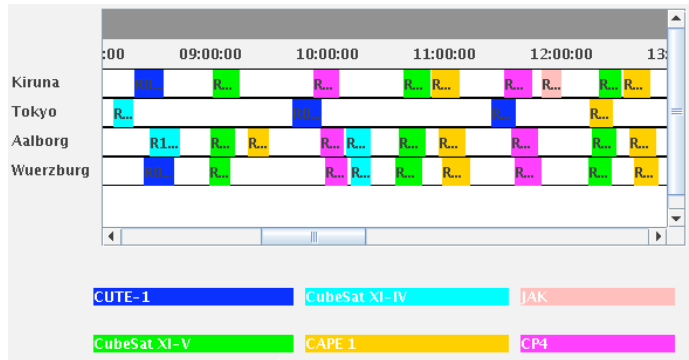

Figure 3: Schedule for the scenario

Conclusion

In this paper a scheduling system for academic ground station networks with redundant scheduling capabilities was introduced. The mathematical definition of a cost function for fair distribution of redundant contact windows deals with scheduling scenarios from the RRSS problem space. The presented approach fits better in the scope of non-commercial ground station networks, as it better satisfies the redundancy and flexibility requirements of those networks. The first results promise great potential for future ground station networks to increase the utilization.
Future work will analyze in more detail the influence of redundancy on the scheduling problem of academic ground station networks. Furthermore the flexibility aspect will be elaborated more with respect to the cost function.

References

[Barbulescu et al. 2002] Barbulescu, L.; A. Howe, A.; Watson, J.; and Whitley, D. 2002. Satellite range scheduling: A comparison of genetic, heuristic and local search. In *Seventh International Conference on Parallel Problem Solving from Nature*.

[Barbulescu et al. 2004] Barbulescu, L.; Watson, J.; Whitley, D.; and Howe, A. 2004. Scheduling space-ground communication for the air force satellite control network. *Journal of Scheduling* 7(1):7–34.

[Barbulescu et al. 2007] Barbulescu, L.; Howe, A.; Whitley, D.; and Roberts, M. 2007. Understanding algorithm performance on an oversubscribed scheduling application. In *Conference on Artificial Intelligence AAAI*, 577–615.

[Barbulescu, Kramer, and Smith 2007] Barbulescu, L.; Kramer, L.; and Smith, S. 2007. Benchmark problems

for oversubscribed scheduling. In *The 17th International Conference on Automated Planning & Scheduling*.

[Damiani et al. 2007] Damiani, S.; Dreihahn, H.; Noll, J.; Niezette, M.; and Calzolari, G. P. 2007. A planning and scheduling system to allocate esa ground station network services. In *ICAPS*.

[Globus et al. 2004] Globus, A.; Crawford, J.; Lohn, J.; and Pryor, A. 2004. A comparison of techniques for scheduling earth observing satellites. In *16th Conference on the Innovative Applications of Artificial Intelligence*.

[Nakamura and Nakasuka 2006] Nakamura, Y., and Nakasuka, S. 2006. Ground station networks to improve operations efficiency of small satellites and its operation scheduling method. In *IAC*.

[Schalck 1993] Schalck, M. 1993. Automating satellite range scheduling. Master's thesis, Graduate School of Engineering of the Air Force Institute of Technology.

[Schmidt and Schilling 2008] Schmidt, M., and Schilling, K. 2008. Satellite scheduling for educational ground station networks. In *International Astronautical Congress, Glasgow*.

[Schmidt, Rybysc, and Schilling 2008] Schmidt, M.; Rybysc, M.; and Schilling, K. 2008. A scheduling algorithm for ground station networks related to small satellites. In *SpaceOps*.

[Schmidt 2008] Schmidt, M. 2008. Extended scheduling function for equal distribution of redundancy. Technical report, University of Wuerzburg.

[Shirville and Klofas 2007] Shirville, G., and Klofas, B. 2007. Genso: A global ground station network. In *AMSAT Symposium*.