Commentary on: Managing SOHO's Keyhole Periods: Problem Definition and Solving Model

Authors: Nicola Policella, Henrique Oliveira Tero Siiliby

Commentator: Russell Knight, JPL

This paper describes computing downlink schedules to manage data flow for the SOHO mission by transforming the data management problem into a flow network problem.

In general, this is a well-written paper that describes a real-world integration. Understandable simplifications are made with respect to software interface (file interfaces) so that integration into existing product flows can be seamlessly achieved. Also, the technique of characterizing data flow management as a network flow problem seems elegant and quite applicable.

In the abstract, the requirement of "Robustness against communications losses and personnel resources" is claimed as a goal, but it isn't clear how this robustness is achieved. Certainly, requiring less in terms of personnel resources is a form of robustness, but usually this would be couched in terms of efficiency or cost reduction.

Also, the problem description claims that the task at hand is to decide "which instrument's data are recorded and when." But, the network flow might decide to split data across nodes representing downlinks and carry data forward to the next downlink, even if a previous downlink had not been completely utilized. Is it the case that these were ignored as artifacts and all available data was downlinked if possible?

Also, what happens when instrument data from one instrument is of higher priority than another? How would this be handled? Note that other systems use linear program formulations for such mixed objective data flow problems.

Finally, similar problems with a few more constraints, e.g., data collection (imaging) that can not be broken up into smaller pieces, are NP-hard. Are there any real-world constraints that cause this problem to become harder? E.g, minimum communication window duration, minimum data block size, multiple download priorities, etc. If there are, do these make the problem NP-hard? If any do, what is the error of approximation?