SOLVING OF SITUATIONAL PROBLEMS HAVING EQUAL SITUATIONAL CONDITIONS

Atanas Atanassov

Space Research and Technology Institute – Bulgarian Academy of Sciences
e-mail:At_M_Atanassov@yahoo.com

Key words: *orbital situational problems; orbital situational conditions, constraints satisfaction problems*

Abstract: *Scientific space mission analysis and design is important process which is necessary and envelopes different stages of space missions preparation and realization. This process begins with an assessment of the possibility of solving a variety of scientific tasks, such as optimization of different satellite orbital parameters and scientific instruments. Reducing/shortening of the time for mission preparation and lowering costs are also important goals [Wertz]. Some elements of SMAD process continue during of mission control - planning and scheduling.*

Each scientific problem, which is solved through satellite data, demands appropriately performed experiments and measurements executed/accomplished in the frames of suitable time intervals under specific constraining conditions of geometrical or physical nature. Determination of these time intervals is achieved through so-called situational analysis. Situational problem solving is connected with checking of situational conditions. Each situational problem contains one or more situational conditions. Verification of some situation conditions may require a significant amount of calculation time. Effective simultaneous solving of many situational problems with difficult for computation conditions could be serious problem due to time for calculation. Optimization of situational analysis algorithms is an irresistible challenge.

Some of situational problems may contain one or more situational conditions heaving same type and equal attributes. The calculation of such situational conditions every time in the frames of different situational problems is inefficiently.

An idea about an algorithm which escapes from such over-calculation of equivalent conditions is submitted in the present report.

РЕШАВАНЕ НА СИТУАЦИОННИ ЗАДАЧИ ИМАЩИ ЕДНАКВИ СИТУАЦИОННИ УСЛОВИЯ

Атанас Атанасов

Институт за космически изследвания и технологии – Българска академия на науките
e-mail:At_M_Atanassov@yahoo.com

Резюме: *Разглежда се възможност за създаване на алгоритъм за решаване на ситуационни задачи с повече ситуационни условия, сред които има еднакви (от един и същи тип и с еднакви параметри) за някои от ситуационните задачи.*

Предлагания подход се основава на обратни отношения, от ситуационните условия към ситуационните задачи. Подходът е универсален и може да се прилага в общия случай на задачи имащи или без еднакви ситуационни условия.

Introduction

Preliminary analysis of a space mission checks the feasibility of planned experiments and measurements in order to obtain quality data to solve the given scientific tasks. Multiple analyses are accomplished to establish optimal values of different orbital and ballistic parameters, positioning the various instruments on satellite platform and parameters of scientific instruments. The goal is to achieve optimal mission results. Appropriate orbit is determined on the first stage of preparation of the mission and test of basic ideas about proposed satellite experiments are investigated [1]. All orbital and scientific instruments parameters could be optimized on the next stages of mission preparation. Recently, requirements have been laid for quick preparation of cheap space missions [1, 2]. This is possible using appropriate flexible and portable new satellite missions software tools for computer simulation and analysis.

Some power commercial tools for space mission analysis and design are known [1, 3, 4] as well as other which are shareware [5, 6, 7].

The execution of various satellite operations related to solving of scientific problems, as well as control and support of different kinds of auxiliary systems, is subject to various restrictive conditions. The determination of the time intervals in which the restrictive conditions are in line with the requirements for performing satellite operations is based on a relevant analysis known as a situational analysis [8, 9, 10].

Computer program for situational problem solving for astrophysical experiment in the frame of second bulgarian cosmonaut mission was developed [11]. In this ‘ad-hock’ created program many situational problems (related to hundreds astrophysical objects) compiled from several situational conditions and corresponding constraints were solved. These conditions was related to transition of Mir station through Earth’s shadow, angles between objects of interest according to Sun and Moon, radiative background and some other. All of situational problems were identical, with only different coordinates of the astrophysical objects of interest. The program was not flexible concerning to mixing situational problems with different type and number of situational conditions.

Approach for situational analysis which copes with "over-calculation problem" when different situational problems have equal situational conditions is presented in the article.

State of the problem

The situational problem (SP) can be presented as complex logical function in the general case.

1. $S = S(\vec{R}, A, t) = \overline{0,1}$

In (1) $\{\vec{R}\} = \langle \vec{r}_1(t), \vec{r}_2(t), \ldots, \vec{r}_n(t) \rangle$ is set of the radii-vectors of the objects in the model space, $\{A\} = \langle \alpha_1(t), \alpha_2(t), \ldots, \alpha_m(t) \rangle$ - set of vector or scalar fields, describing certain properties of the model space and t – the time.

Actually, we can have a combination of several constraining situational conditions (SC) for one situation problem. In addition, we shall examine such conditions that are independent from each other. Thus, the set of situation conditions $\{\gamma_i\}$ can be juxtaposed to the set of predicate functions $\{s_i\}$.

The presence of situation event S will require the fulfillment of the following identity:

2. $S = s_1 \wedge s_2 \wedge \ldots \wedge s_n = 1$

In the field of space investigations, one situation event occurs when S = 1. All predicate functions in (2) are justified in time interval (t_{final} - t_{start}).

There are different types of situational conditions:

- **geometrical** - visibility and lightening of objects of interest, angles between them and sources of light (Sun and Moon);
- **physical** - related to values of physical quantities (fields) and orientation according to vector fields (like magnetic field);
- **access to shared resources** - related to auxiliary apparatus and systems of the satellite (storage, communication tools).

Most often, a complex of specific situational conditions that meet specific constraints is required for performing a scientific or technological task. This happens at a certain time interval of some part of the orbit. The duration of the time interval, when the situation problem is executed (the value of the predicate function has a value of 1) may also be important for the execution of the satellite operations.

Let us pay attention that the planning and scheduling of satellite operations is a next stage that is dependent on the possible, previously determined time intervals. The situational analysis examines only part of the required conditions for carrying out various operations, related to solving different scientific tasks and determines the timeframes optimal for execution. A schedule shall be drawn up, which takes account of the need for joint use of different resources for the purposes of different scientific and technological tasks, for the implementation of requested satellite operations. Different satellite operations (similar operations are possible for different scientific tasks) compete for the use of common resources. A schedule for the implementation of satellite operations would be prepared, which takes into account the need of shared use of resources for the purposes of different scientific and technological tasks. Different algorithms and strategies to achieve schedules are used [12, 13, 14]. For the optimal use of satellite systems, this analysis uses different algorithms and strategies to achieve close to optimal schedules.

Previous work

Initially, a model for situational problems definition was developed [15]. Each situational problem is presented trough structured data (user defined type). Such data types for each situational condition are saved in library- fortran module file. Every type contains specific attributes characterizing respective situational condition. Each situational problem is composed from several situational conditions. Dialog editor for assembling situational problems by selection of situational conditions and setting of their attributes.

Serial [15] and parallel [16] versions of situational problem solver were developed. The situational problems are treated as independent. If two situational problems contain equal situational conditions, they are checked in consecutive order every time.

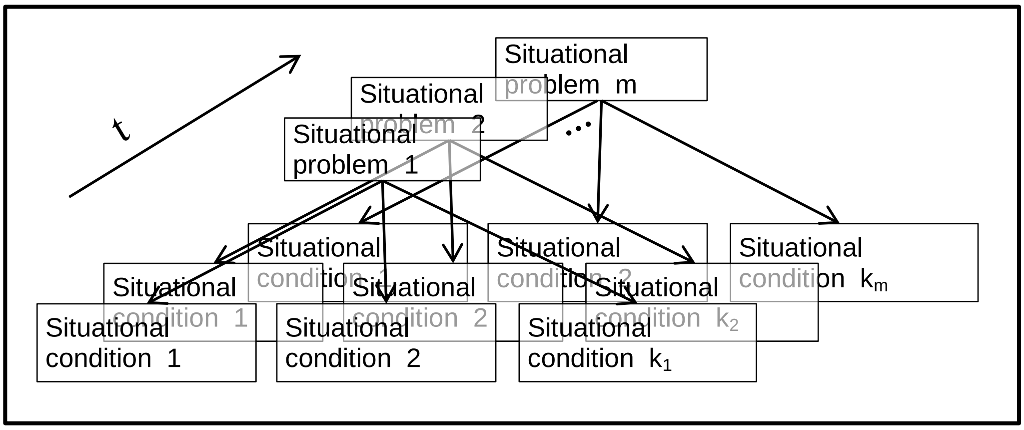

Fig. 1. All of situational problems are checked in consecutive order in the time

Setting of the problem

Let S_1 and S_2 are two situation problems containing more than one situational condition - N and M respectively:

$$S_1 = s_{1,1} \wedge s_{1,2} \ldots s_{1,N} \equiv \bigwedge_{i=1,N} s_{1,i}$$

$$S_2 = s_{2,1} \wedge s_{2,2} \ldots s_{2,M} \equiv \bigwedge_{j=1,M} s_{2,j}$$

If for some i and j the situational conditions are of one type and have the same attributes, a situational task will have equal situational conditions. Two situational problems can have more than one equal situational condition. The above condition from mathematical point of view could be written down:

$$\exists \left\langle \mathrm{i}, \mathrm{j} \right\rangle_{\mathrm{m};1\le \mathrm{m}<\min(\mathrm{N},\mathrm{M})},\ \mathrm{s}_{1,\mathrm{i};\mathrm{m}} = \mathrm{s}_{2,\mathrm{j};\mathrm{m}}.$$

If the situational problems are solved one by one (fig. 1), without accounting values of already calculated predicate functions (situational conditions) from previously solved situational problems, over-calculations could be produced. Let have in mind that these unnecessary calculations are repeated on each step of simulation time. To find approach for minimization of these calculations is important.

Proposed algorithm

Optimizing situational analysis, when some of the situational conditions of different situational problems are equal is possible, based on the following approach.

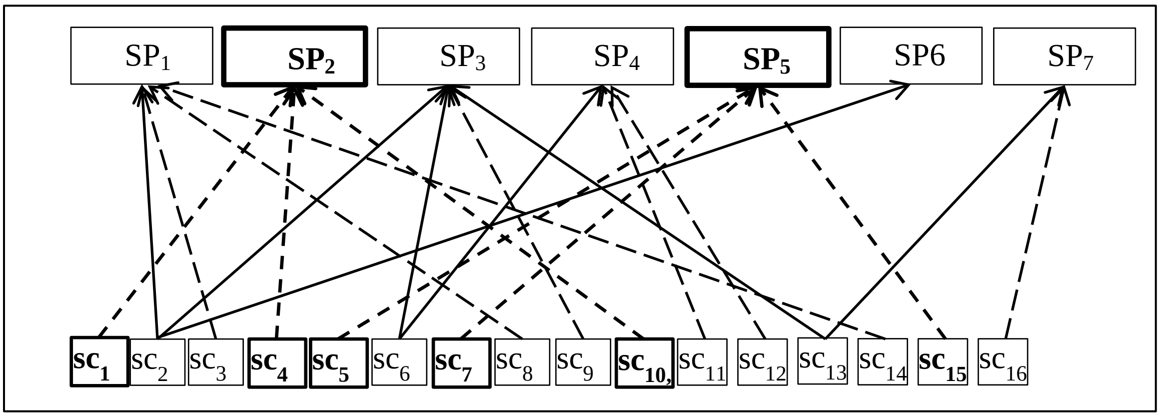

Fig. 2 The series of rectangles SP and sc illustrate the queues of situational problems and situational conditions respectively. Rectangles drawn with thick lines represent situational problems (**SP_2** and **SP_5**) having different situational conditions – (sc_1, sc_4, sc_{10} and sc5, sc7, sc_{15}). Other situational problems (SP_1, SP_3, SP_4, SP_6, SP_7) have mixed situational conditions- some of them are equal (sc_2, sc_6, sc_{13}) and other not.

At the beginning we have set of situational problems and relations between each of them and correspondent situational conditions. The proposed approach is based on reversed relations between each situational condition and all situational problems in which it participates. A dynamic storage/array is created which contains the names (identification code) of all situational conditions. Except these names this array contains and three additional attributes for each of situational conditions - number of situational problems where it participates, initial address of the storage where identification codes of these situational problems are dynamical allocated and series of identification numbers of SP.

Two queues, one containing all situational conditions (SP_Q) and other situational problems (SC_Q) are prepared before the beginning of simulation process (the simulation process contains situational analysis). All situational conditions and situational problems are arranged in respective queue in random order or according to the order of their definition.

When situational tasks are solved, situational conditions are checked in the order they are queued. The corresponding situational problems are marked as false or are deleted in the situational problems queue (SP_Q), when a certain situational condition is not fulfilled.

```
   CALL  Making_Queue(SP_Q)
     CALL  Making_Queue(SC_Q)
  …
DO WHILE(SC_Q ≠ empty)
        iSC= check_prFunc(SC_Q; <parameters>) ! predicate function (situational condition) checking
    IF(iSC   .EQ.   .false.) THEN ! Not satisfaction of situational condition
           kod=  SP_reject(SP_Q) ! corresponded situational problems are rejected-
                                 ! they will not been satisficed
           kod= SC_reject(SC_Q) ! rejects situational conditions of unfulfilled situational problems
       ELSEIF(iSC   .EQ.   .true.) THEN
      ENDIF
END DO
 CALL  sit_intervals(SP_Q) ! Time intervals determination
  …
```

Fig. 3. Code fragment illustrated the presented algorithm

Other situational conditions located further in the queue and associated with the same situational problem may remain unchecked after a currently tested unfulfillable situation condition (the value of the predicate function is 0 and the situational problem is deleted). If these situational conditions continue to stay in the queue (SC_Q) for following checking this will produce unnecessary, over calculations.

Copping with this issue is possible by two ways. The first one is related to preliminary check if the situational problem is in the queue of the problems. According to the second approach, all situational conditions related to the particular unfilled situational problem are deleted. After that, a checking of the next situational condition in the queue is performed.

When all situation conditions in the SC_Q queue are finished (all correspond predicate functions are checked), in the queue of situational problems SP_Q may eventually remain only such tasks, that can be executable according to the constraints in situational conditions at all current points in the time. The entire time interval when a situational problem is executable is determined on the basis of all consecutive moments of time when the relevant conditions are met.

Conclusion and future work

The proposed algorithm for situational analysis is universal unlike the previously developed one [16]. This algorithm could be applied to solve situational problems no matter whether they have or not equal situational conditions. A case when all situational problems haven't equal situational conditions is possible. So, this approach is universal and is interesting to investigate and analyze its priorities and disadvantages in different cases. Experiments for comparison of efficiency of the two algorithms are necessary to be performed.

Special attention will be directed toward establishing potential possibilities for parallelization of this algorithm. The influence of data locality and latency of different storage levels will be investigated too. The last is important when high-dimensional problems are solved.

Reference:

1. Wertz, J.R., W.J. Larson, Space Mission Analysis and Design, third ed. Microcosm Press, Kluwer Academic Publishers, 1999.
2. Trivailo, Olga, Martin Sippel, and Y. Ahmet Şekercioğlu. "Review of hardware cost estimation methods, models and tools applied to early phases of space mission planning." *Progress in Aerospace Sciences* 53, 2012, 1-17.
3. https://www.agi.com/products/engineering-tools
4. https://ai-solutions.com/freeflyer/
5. Hughes, Steve. "General Mission Analysis Tool (GMAT) Mathematical Specifications." (2007).
6. Hughes, Steven P., et al. "Verification and validation of the general mission analysis tool (GMAT)." (2014).
7. Qureshi, Rizwan H., and Steven P. Hughes. "Preparing general mission analysis tool for operational maneuver planning of the advanced composition explorer mission." *AIAA/AAS Astrodynamics Specialist Conference, San Diego, CA*. 2014.
8. Prokhorenko, V. I. "Study of satellite situations mission." *Acta Astronautica* 10.7 (1983): 499-503.
9. Nazirov, R. R., and V. I. Prokhorenko. "Situation analysis in the problems of space physics." *Kosmicheskie Issledovaniia* 36 (1998): 311-322.
10. Prokhorenko, V. "Mission analysis for the INTERBALL project. Pre-launch orbit selection and long-term experiments planning." *INTERBALL mission and payload*, 46, 1995.
11. Gaidarov, P., Sukhanov, A., Atanasssov, A., Rjazanova, V., Planirovanie astrofizicheskih eksperimentov v proekte "Shipka", Pr.- 1534, IKI AN SSSR, 1989.
12. Pemberton, J. C. "Towards scheduling over-constrained remote sensing satellites." *Proceedings of the 2d International Workshop on Planning and Scheduling for Space*. 2000.
13. Pemberton, Joseph C., and Flavius Galiber. "A constraint-based approach to satellite scheduling." *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* 57, 101-114, 2001.
14. Barbulescu, Laura, et al. "Scheduling space–ground communications for the air force satellite control network." *Journal of Scheduling* 7.1, 7-34, 2004.
15. Atanassov, Atanas. "Program System for Space Missions Simulation–First Stages of Projecting and Realization." *Proceedings of SES* 2013, 209-214, 2012.
16. Atanassov, Atanas Marinov. "Parallel satellite orbital situational problems solver for space missions design and control." *Advances in Space Research* 58.9, 1819-1826, 2016.