Bulgarian Academy of Sciences. Space Research and Technology Institute.
Aerospace Research in Bulgaria. 38, 2026, Sofia

DOI: https://doi.org/10.3897/arb.v38.e01

APPLYING REORDERED HEURISTICS FOR EFFICIENCY ENHANCEMENT IN SITUATIONAL ANALYSIS WHEN SOLVING PROBLEMS CONTAINING MORE THAN ONE CONDITION IN SPACE MISSION ANALYSYS

Atanas Atanassov

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

Key words: *situational analysis, reordered heuristics, space mission analysis and design*, *orbital events prediction*

Abstract

Situational analysis deals with verifying conditions of a geometric or physical nature during satellites' movement in their orbits. This analysis is applied to determine time intervals optimal for the execution of specific satellite operations. Situational problems are solved at various stages of the preparation and operational phases of the satellite missions. The most interesting cases are when a situational problem contains multiple conditions. Each situational problem is solved by sequential verifications of the conditions involved. Verifications can be terminated when an unfulfilled condition is detected (Horner's rule) to improve the efficiency of the analysis. Calculations are repeated for subsequent time steps. Solving such a situational problem may be time-consuming when multi-variant and multi-satellite simulations are performed.

A heuristic approach is proposed for the efficiency enhancement of situational analysis for problems including multiple conditions. This approach is based on the reordering of condition checks. The approach is algorithmized and incorporated into the situational analysis solver that was previously developed. The efficiency of the proposed heuristic is evaluated based on examples with different numbers of situational conditions. This approach shows better efficiency than applying Horner's rule.

Introduction

The rapid advancement of space technology has led to a significant increase in the spacecraft population orbiting the Earth. Furthermore, satellites are becoming increasingly smaller. Multi-satellite missions are designed and implemented in various scientific and applied fields. These and other trends are well-represented in the scientific literature [1, 2].

Limitations are imposed on the preparation time, cost, and reliability of operational use when designing and preparing space missions, regardless of the number and type of satellites. Computer programs are used at different stages from the preparation to the realisation of space missions. Various aspects of space missions are simulated to meet the ever-increasing requirements. It is important to note that, in addition to well-known programs designed for commercial distribution (STK[1], FreeFlyer[2], ASTOS[3] – or open source program GMAT[4]), some organisations develop their own programs to meet specific needs. Some of these internal programs extend beyond satellite dynamics simulations [3–6]. Wertz & Larson [7] suggest using programs tested in numerous missions created and supported by specialised companies. Nonetheless, he highlights that developing one's own programs allows for a deeper understanding of the underlying problems. Chagas [8] emphasises the importance of applying a comprehensive set of modelling and simulation toolkits to develop complex space missions. Applying multi-physics environments is crucial to tame the complex problems in space mission design [9].

The situational analysis deals with the determination of time intervals optimal for the implementation of certain satellite operations (performing measurements with various sensors, communication with ground stations, and special or general satellite operations). The fulfilment of specific conditions is verified for separate moments throughout the simulation time in the framework of this analysis. This analysis examines the environmental conditions under which satellites operate. We assume that the corresponding orbital event occurs when one or a combination of several conditions is met. Situational analysis, along with orbital analysis, satellite operations planning, and various engineering/mechatronic-oriented analyses, plays a critical role in optimising satellite missions. Applying this analysis is crucial at different stages of the preparation and execution of space missions.

At the Space Research and Technology Institute, BAS, work is being done on creating and improving the Parallel Situational Analysis Solver (PSAS) [10]. A reordered heuristic applicable to solving situational problems with several conditions is proposed in the present work.

The rest of the paper is organized as follows: The second section describes some types of missions and the specific situational conditions used in each of them; The third section presents the idea of situational problems composed of several conditions and their solution; The fourth section explains the approach for efficiency enhancement; The fifth section tells about parallel situation analysis solver; The sixth section shows application examples and an estimation of the efficiency of the proposed heuristic; Some comments about algorithmic complexity are presented in the seventh section and the eighth section concludes the article and indicates some possible developments and applications.

---

[1] STK [Online]. Available at: https://www.ansys.com/content/dam/amp/2023/november/asset-creation/stk-capabilities.pdf
[2] FreeFlyer [Online]. Available at: https://ai-solutions.com/freeflyer-astrodynamic-software/
[3] ASTOS [Online]. Available at: https://www.astos.de/products/astos
[4] GMAT [Online]. Available at: https://gmat.atlassian.net/wiki/spaces/GW/overview

Related Work

Determining the optimal time intervals for observation and measurement is crucial at the initial stage of space mission preparation. Performing a situational analysis is indispensable for assessing the feasibility and realism of any mission. At the next stages, parameters related to various satellite subsystems and instruments can be optimised. These time intervals are used as input resources when planning and scheduling payload operations at the operation stage of a satellite mission [11].

The ability to make observations and measurements depends on the platform's location in orbit and various other conditions that either allow or prohibit them. These conditions are related to the visibility of the objects being monitored (or from the disposition of in-situ measurement instruments to be in suitable spatial areas) and any sources of disturbance or prohibitions caused by other objects or phenomena.

Forecasting the optimal time intervals for observations in the frame of astrophysical missions necessitates the assessment of various factors, some of them related to the positions of the Sun, Moon, and satellite platform [12–17]. Additionally, physical factors such as the satellite's position relative to the South Atlantic Anomaly and polar cusps must also be considered [16, 18]. The presence of high levels of radiation background in certain segments of the orbits above these areas is a significant issue, as it can lead to electrical discharges, scientific instrument damage, and background enhancement in images obtained by optical instruments with SSD detectors. Constraints related to passing through radiation belt passages regarding XMM-Newton[5] and Chandra missions are applied [19]. Thus, an analysis of geometrical and physical factors is essential when selecting the appropriate observation intervals, especially when dealing with sensitive scientific instruments and image acquisition systems. Bernacca et al. [20] present examples of how to simulate star visibility and tracking by a telescope attached to the main truss of the International Space Station. Additionally, to determine the visibility of the studied stars, solar panels are considered as structural elements in this work.

Interesting and current tasks arise regarding space debris tracking with space-based optical sensors. Scott and Thorsteinson [16] consider different constraints regarding NEOSSat Space-Based space situational *awareness*. Mission representing a microsatellite space telescope designed to track resident space objects and perform asteroid astronomy. Orbital conditions related to constraints related to the satellite platform and its orientation relative to the Sun and the Earth, the observation objects, and the mutual position of the platform and the observed objects (resident space objects) are considered in detail. Further constraints are applied to the telescope's field of view – avoiding the galactic centre and

---

[5] ”XMM-Newton Users Handbook”, Issue 2.22, 2024 (ESA: XMM-Newton SOC). Avalaible on: https://xmm-tools.cosmos.esa.int/external/xmm_user_support/documentation/uhb/XMM_UHB.pdf

containing background stars. Du et. al. [21] and Du et. al. [22] presented results from computer simulations to evaluate the performance of multi-satellite space debris cataloguing missions. In both cases, in addition to geometric constraints to estimate visibility, an illuminance estimate is added, based on the shape and albedo of the space debris. Li et al. [23] apply the same constraints in their analysis.

Experimental observations of the anomalous fluxes of fast charged particles, disturbances of quasi-steady and ULF–VLF magnetic field components, and thermal plasma ion fluxes have been considered using the data of the active space experiment during electron beam injection in the frame of the Apex mission [24]. A DOK-S charged particle spectrometer installed at the subsatellite measured the differential fluxes of fast electrons and protons in eight energy intervals. The conditions for executing the experiment are based on the angle between the injection direction and the magnetic field vector, the characteristics of the environment, as well as the possibilities of observations with instruments located on a sub-satellite or on the Earth's surface.

In magnetospheric research missions, it is necessary to know when the satellites pass through the different regions (polar cusp, polar oval, bow shock, or/and magnetopause) for preliminary analysis and measurement planning. These regions are described with geometric models of surfaces [25] and determine the conditions for the operation of the instruments.

The visibility of objects and areas of the Earth's surface depends on the parameters of the Earth's orbit and its rotation in remote sensing missions. The ability to observe the Earth's surface depends on the spectral ranges of instruments. For observations in the visible spectral range, the objects must be illuminated by the Sun. Observations of the unilluminated part of the Earth's surface are carried out in the infrared spectral range, as the satellite may be in the umbra [26]. Kosari et al. [27] involve additional considerations of conditions related to pixel elongation and lighting (concerning the Sun) constraints.

Within the framework of the Korean space program, tools are being developed to predict orbital events [3, 28] – satellite contact time in specific ground stations, eclipses due to the Earth and moon, satellite pass time at the equator, pole, apogee, and perigee, satellite earth sensor interferences due to the sun and moon, Satellite attitude manoeuvre for imaging.

Concept of situation analysis

As pointed out above, situational analysis is applied to the space mission analysis at the initial stages of its preparation. During the operational stage, this analysis provides planning and scheduling for satellite operations, with appropriate time intervals for their execution. Each situational problem SP is composed of one or a conjunction of several situational conditions sc_i of different types:

(1) $$SP = sc_1 \wedge sc_2 \wedge \ldots \wedge sc_n = \bigwedge_1^n sc_i,$$

Applying Horner's rule, we can write

(2) $$SP = (\dots(\dots(sc_1 \wedge sc_2)\wedge \dots)\wedge sc_m \dots \wedge sc_{n-1})\wedge sc_n = \bigwedge_1^m sc_i \,, \ (m \leq n)$$

where m is the index of the first unfulfilled condition among the ordered n conditions of the situational problem. While the algorithmic complexity of representation Eq. (1) is $O(n)$, that of Eq. (2) is $O(m)$ $(m \leq n)$. The conditions themselves are predicate functions accepting values of ***true*** or ***false***. They can be generally represented as:

$$sc_i = sc_i\left(\{\vec{R}\}, \{\alpha\}, \{\beta\}, t\right)$$

where $\{\vec{R}\}$ is a set of radius vectors of satellites, $\{\alpha\}$ – is a set of parameters of the mathematical model describing the situational condition, and $\{\beta\}$ – is a set of specific constraints. The sets of parameters $\{\alpha\}$ and constraints $\{\beta\}$ are specific for each situational condition. In some cases, they are explicitly entered when composing the situational task (conditions 1 and 4 in Table 1). They are defined by models in other cases. This is the case of conditions 2 and 3 in Table 1. The radius vectors of the Sun and Moon are parameters of the geometric models of the corresponding situational conditions, which are calculated based on models of their movements. In this formulation, solving a situational problem represents a verification of the feasibility of a logical function at appropriate moments.

Enhancement of situational analysis efficiency

Basics

Improving the effectiveness of situational analysis is important for the following reasons:

- A large number of situational problems must be solved; the large number of problems is a result of the number of satellites, scientific instruments, objects to study, and scientific tasks to solve;
- Solving problems with several situational conditions each requires more calculations;
- Multiple solving while varying different parameters (orbital parameters, parameters of satellite subsystems) in order to optimise them;
- Running computer simulations over a long time horizon.

There are some standard ways to increase the efficiency of situational analysis:

- Optimisation related to mathematical models for verifying situational conditions;
- Use of compilers generating efficient code;
- Application of parallel calculations.

In addition to the previously mentioned ways for enhancing the efficiency of situational analysis, we will focus on one more.

Enhancing the efficiency of situational analysis by reordering the heuristic

Heuristics are ways of solving problems based on practical or intuitive assumptions about optimality. Unlike methods, they are not associated with rigorous evidence. Here, a heuristic to enhance the efficiency of solving multi-condition situational problems by reordering them is proposed – the reordered heuristic.

Instead of the direct problem SP to verify Eq. (1) or Eq. (2), let us consider the opposite problem [29]:

(3) $$\overline{SP} = \overline{sc(t)_1 \wedge sc(t)_2 \wedge \ldots \wedge sc(t)_n}$$

According to (1) or (2), we look for time intervals when all conditions are met. In the opposite task (3), we look for time intervals when at least one of the conditions is not fulfilled. Applying de Morgan's law, we can represent the opposite problem Eq. (3) as follows

(3') $$\overline{SP} = \overline{sc(t)_1} \vee \overline{sc(t)_2} \vee \ldots \vee \overline{sc(t)_k} \vee \ldots \vee \overline{sc(t)_n}$$

To solve the opposite problem $\overline{SP}$it is enough that only one of the conditions $\overline{sc(t)_k}$ is fulfilled. After finding the k^{-th} condition ($\overline{sc(t)_k} = true$) sufficient to solve the problem, one can proceed with verifications in the next steps by applying the following heuristic:

(4) $$\overline{SP} = \overline{sc(t+\triangle t)_k} \vee \overline{sc(t+\triangle t)_1} \vee \ldots \vee \overline{sc(t+\triangle t)_{k-1}} \vee \overline{sc(t+\triangle t)_{k+1}} \vee \ldots \\ \ldots \vee \overline{sc(t+\triangle t)_n}$$

The meaning of reordering the conditions is related to the fact that the situational conditions themselves are slowly changing functions with respect to the time step with which the simulation model is discretised. In fact, moving the k^{-th} unfulfilled condition in the first place causes verifications to interrupt already on the very first condition for a large number of time steps.

Development of a situational analysis solver

Situational problems editor

A dialogue editor is developed for composing situational problems. Depending on the type of satellite mission, a set of possible situational conditions is proposed to choose from. With appropriate dialogue controls, situational conditions are selected, and their parameters are entered to compose a situational problem. With check control, it is indicated that the current problem will be solved without or based on a reordered heuristic.

After compiling all situational problems, an actual solver is created to solve them at each step of the system/simulation time. For this purpose, a special subroutine is used, which in turn calls system functions to create computational threads. In addition, two objects of type event are created for each thread. These events are used for synchronisation between the solver threads and a special thread (parent thread) that manages the simulation process. Here, we briefly note that the entire simulation process can include different types of parallel solvers. The handlers of the threads and the mentioned events are stored in a two-dimensional array and represent pool parameters.

When solving numerous situational problems (hundreds and thousands) it is important to avoid overload. Managing the thread pool consumes computational resources; therefore, taking only one task is inefficient. For this purpose, an integer variable is created – “granularity”. It determines the number of situational problems that are taken by the threads to solve at once. When these problems are solved, the thread can try to take the next portion of the problems. One more object of type “event” is created for all solver threads. It is used by each thread when solving several situational problems (the number is specified with the parameter "granularity"). When a thread needs to take on situational problems for solving, it changes the state of this event and takes certain actions. Until these actions are completed and the thread returns the previous state of the event, access to the situational problems by other threads is forbidden. This prevents situational problems from being solved by different threads. A situational problem counter is also necessary for the thread pool to function. It counts the situational problems taken for solving.

In addition to the situational problem descriptor, the coordinates of the satellites are also needed to solve situational problems. Depending on the type of situational conditions, the coordinates of other objects or environmental parameters may be required. All necessary data, or their memory addresses, are obtained from the Space mission dynamic solver descriptor (SMDSD) (Fig. 1a). This descriptor is necessary for the interpreter of the simulation model to start the situational solver at each time step. This is shown in Fig. 1b with the bidirectional connections between "Interpretation thread of a simulation model" and “pool of threads control” subroutine. The descriptor contains heterogeneous information about the pull of the thread of the solver and situational problems. Information from this descriptor is used when creating functional models of satellite subsystems.

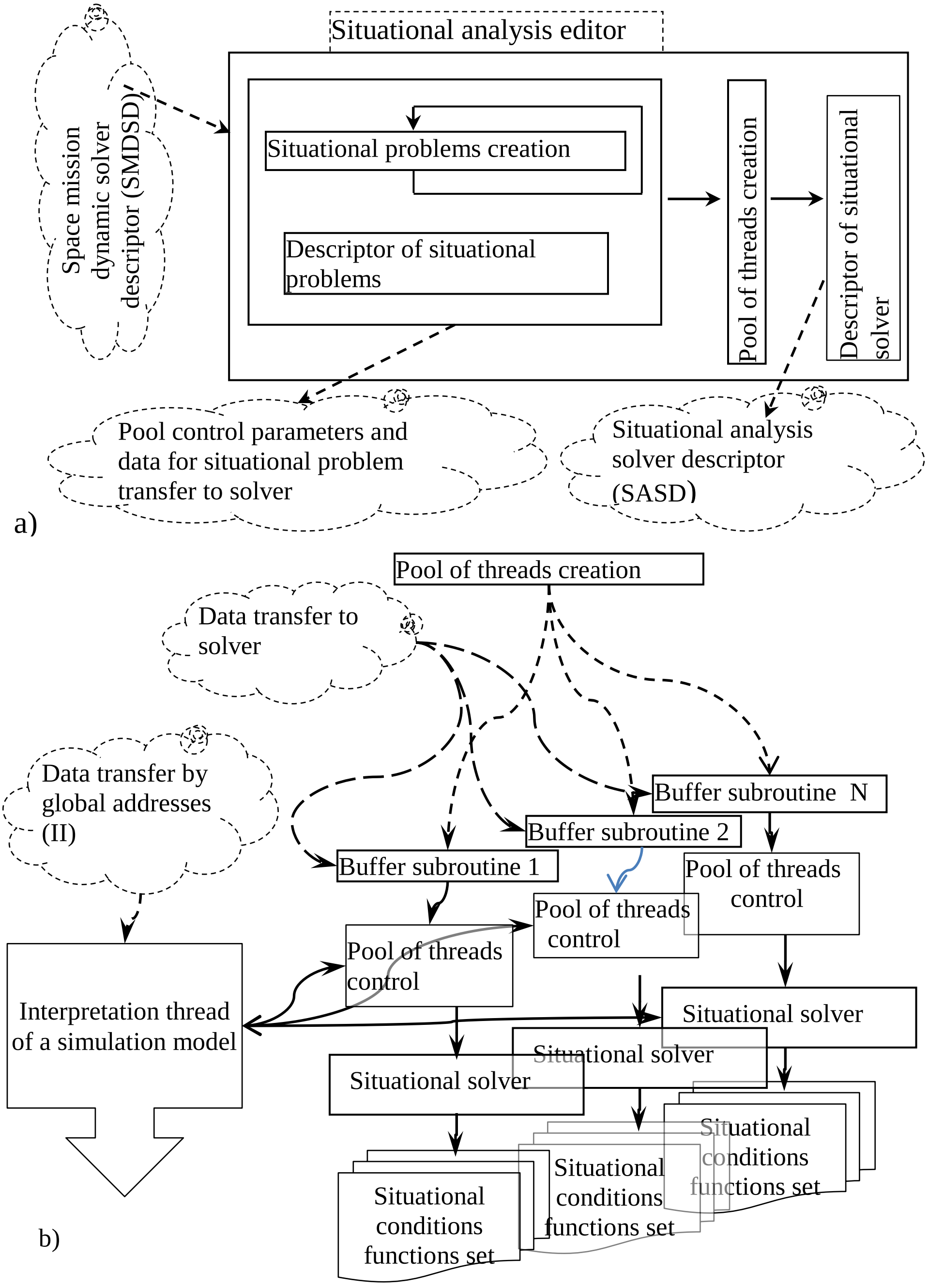

Fig. 1. Structural schemes of: (a) Situational Problems Editor and (b) Situational Problems Solver composed of N threads

Parallel situational analysis solver

A parallel solver of situational problems was developed [10]. The "thread pool" model is used for parallelisation [30]. “Parallelism across the situational problem” is applied as the parallelisation approach. The solver threads are synchronised for each time step with the parent thread (Fig. 1b). Depending on the particular problems, the number of threads is selected, as well as the “granularity”, i.e., the degree of fragmentation of the entire set of situational problems. The number of threads for each individual solver should conform to the number of processors/cores in the computer system. Within a simulation model, different actual solvers (created with different parameters) can be created to solve specific sets of problems.

Each solver receives the necessary control parameters and data via a buffer subroutine. This buffer subroutine is the main one for the solver. It accepts the data from the situational editor (mentioned above) via global data (common block in the Fortran language) and then calls a thread pool supervising (management) subroutine. Each buffer subroutine contains a specific name of the main subroutine used in a particular solver. The thread pool management subroutine has polymorphic properties and can be called from buffer subroutines of a variety of parallel solvers.

Situational problem descriptor

Each situational problem is described by a descriptor – a one-dimensional array of a derived type (Atanassov, 2016). The null element of the array contains information about the situational problem (Appendix A). The following elements describe the situational conditions, according to the order of their input by the situational editor. The descriptors of all situational problems related to a specific space (single-satellite or multi-satellite) mission are combined into a two-dimensional array. The **efficiency** parameter, which indicates the efficiency-enhancing approach in solving each situational problem is included in the null element (Appendix A). The value of this parameter is determined at the stage of composing the situational problem. At a value of 0, Horner's rule applies. A reordered heuristic is used if the value of 1 is set. Other values of the **efficiency** parameter can be used for other algorithms in the future.

Program implementation of the reordered heuristic

When an unfulfilled condition included in a situational problem is detected, subroutine **Conditions_reordering** is called (Fig. 2). This subroutine reorders some elements in the task descriptor. The null element does not participate in the reordering. The element matching the detected unfulfilled condition is moved to the first position, and all others before it are shifted to one position. The remaining elements do not change their original positions.

```
SUBROUTINE     Conditions_reordering (iadr_sci_task,  max_cond,no_meet)
 USE  DFlib
 USE  RN,ONLY:SitCond,sit_task
   type    (sit_task)           sci_task(0:max_cond)
   type    (SitCond)    Sit_Cond,temp_Sit_Cond(no_meet-1)
 POINTER(iadr_sci_task,sci_task)
 AUTOMATIC  Sit_Cond,temp_Sit_Cond

      IF(sci_task(0)%opt_level.EQ.1)  THEN ! reordered heuristic
         IF(no_meet.EQ.1) RETURN
               Sit_Cond= sci_task(no_meet)%sit_cond
           DO    i=no_meet,2,-1
               sci_task(i)%sit_cond= sci_task(i-1)%sit_cond
           END DO
               sci_task(1)%sit_cond= Sit_Cond
  ELSEIF(sci_task(0)%opt_level.EQ.2)  THEN ! other heuristic
            IF(no_meet.EQ.1) RETURN
   ENDIF

END SUBROUTINE  optimization
```

Fig. 2. The "optimisation" subroutine implementing the "reordered heuristic"

Examples of applications

Situational problem compilation

Let's consider a situational problem composed of several types of conditions (Table 1):

Table 1. Example of a situational problem

Numerical identifier	Titles of the types of situational conditions that are used below
1	The satellite is in the Earth’s shadow
2	The Moon is below the horizon
3	The satellite passes over a circular area of Earth's surface with geographic coordinates of its centre (φ,λ) and angular radius δ.
4	A circular area on the Earth's surface (with geographic coordinates of its centre (φ,λ) and angular radius δ) is fully illuminated

Such situational problems arise in remote sensing missions. The situational conditions mentioned in Table 1 are included in the situational problem editor and the situational solver.

Evaluation of the efficiency depending on the order of the conditions

Three numerical experiments are conducted to evaluate the efficiency of the proposed heuristic. Both experiments simulated the motion of a satellite in an orbit with the following parameters: semi-major axis a = 7 000 000 m, eccentricity e = 0.001, and inclination i of 5°. It is verified whether the satellite passes above a circular area on the surface of the Earth with geographical coordinates of the centre at (0°, 0°), and an angular radius of δ = 10°. Additionally, it is verified whether the satellite is in the shadow of the Earth [31], and if the Moon is below the horizon [32]. The integration of the equations of motion of the satellite and the checks of the conditions are performed within a time horizon of thirty days with a step dt = 10 s.

Table 2 contains results regarding the number of verified conditions when solving the situational problem based on a static ordering of the conditions applying Horner's rule (2) and when applying the reordered heuristic.

Table 2. Number of condition verifications performed – first situational problem

Variants of the conditions ordering	Verified conditions with Horner's role	Verified conditions with heuristic	Efficiency [%]
1-2-3	277 635	263 919	4.94
1-3-2	277 588	263 916	4.92
2-1-3	358 035	263 918	26.29
2-3-1	389 388	263 919	32.22
3-1-2	358 320	263 915	26.35
3-2-1	389 720	263 920	32.28

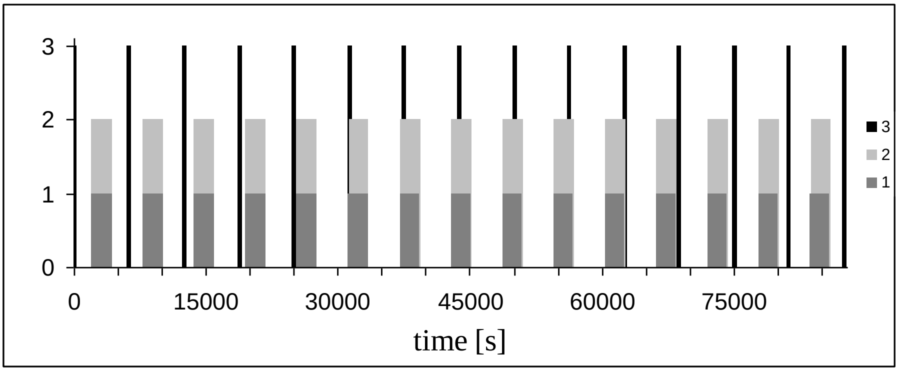

Fig. 3. Time diagrams of situational conditions are shown:
1 – the satellite is in the Earth's shadow, 2 – the satellite is in the shadow of the Moon, and 3 – the satellite passes over a circular region

Fig. 3 shows a timing diagram of the situational conditions for the considered time horizon. The three situational conditions are fulfilled simultaneously in the following time intervals: (31 200, 31 410), (37 340, 37 670), (43 580, 43 920), (49 840, 50 170), (56 100, 56 410), and (62 360, 62 430).

In the second experiment, an additional fourth condition is checked to see if the entire above-mentioned area is unlit by the Sun. In one case, time intervals are searched in which three conditions are met simultaneously, and in the other case, four conditions are met.

Table 3. Number of condition verifications performed – second situational problem

Variants of the conditions ordering	Verified conditions with Horner's role	Verified conditions with heuristic	Efficiency [%]
1-2-3-4	279 574	265 858	4.91
1-2-4-3	282 598	265 859	5.92
1-3-2-4	279 527	265 855	4.89
1-3-4-2	279 945	265 856	5.03
1-4-2-3	284 395	265 854	6.52
1-4-3-2	281 789	265 851	5.66
2-1-3-4	359 974	265 857	26.14
2-1-4-3	362 998	265 858	26.76
2-3-1-4	391 327	265 858	32.06
2-3-4-1	407 843	265 857	34.81
2-4-1-3	405 197	265 855	34.39
2-4-3-1	418 689	265 856	36.50
3-1-2-4	360 259	265 854	26.20
3-1-4-2	360 677	265 855	26.29
3-2-1-4	391 659	265 859	32.12
3-2-4-1	408 175	265 867	34.86
3-4-1-2	403 060	265 852	34.04
3-4-2-1	419 158	265 853	36.57
4-1-2-3	400 503	265 853	33.62
4-1-3-2	397 897	265 850	33.19
4-2-1-3	440 905	265 854	39.70
4-2-3-1	454 397	265 855	41.49
4-3-1-2	438 436	265 851	39.36
4-3-2-1	454 534	265 852	41.51

A simulation experiment was conducted with two satellites. The orbital parameters are as follows: for the first satellite, the semi-major axis is a_1 = 7 000 000 m, eccentricity e_1 = 0.001, and inclination i_1 = 5°; for the second satellite, a_2 = 7 500 000 m, e_2 = 0.005, and i_2 = 15°.

Table 4 shows the results of the checks performed for two cases of initial orders of the situational conditions. The number of possible orders under six conditions is 720 and cannot be shown. In these and other cases, the number of checks based on Horner's rule varies significantly between cases. When applying reordering, much fewer checks are performed, close to those shown in Tables 2 and 3.

Moreover, in this experiment, the conditions are more than those in the second, and given the applied reordered heuristic, the probability of fulfilling more conditions in one time step is smaller than that with fewer conditions. Therefore, the number of checks performed decreases with an increase in the number of conditions when applying this heuristic!

Table 4. The number of checks for problems, including three conditions for each companion for an interval of one month, is shown

Variants of the conditions ordering	Verified conditions with Horner's role	Verified conditions with heuristic	Efficiency [%]
1(1), 2(1), 3(1), 1(2), 2(2), 3(2)	279 637	260 902	6.7
2(1), 3(1), 2(2), 3(2), 1(1), 1(2)	408 989	260 918	36.20

In Table 4 and Table 5, the ordering variants are denoted by two numbers – the first digit indicates the situational condition, while the second digit (in the brackets) refers to the corresponding satellite.

Solving the same situational problem for three ten-day periods of the selected time horizon and results for a randomly chosen order of conditions are shown in Table 5. The differences in the efficiency in solving the situational problem for the three time intervals are due to the irregularities related to changes in the positions of the Sun and Moon.

Table 5. The number of checks performed for the second task from Table 4, but for a consecutive ten-day period

Variants of the conditions ordering	Verified conditions with Horner's role	Verified conditions with heuristic	Efficiency [%]
1-10 Apr 2022			
2(1), 3(1), 2(2), 3(2), 1(1), 1(2)	145 520	86 931	40.26
10-20 Apr 2022			
2(1), 3(1), 2(2), 3(2), 1(1), 1(2)	118 857	86 701	27.05
20-30 Apr 2022			
2(1), 3(1), 2(2), 3(2), 1(1), 1(2)	140 428	86 897	38.1

Effectiveness of the proposed heuristics

Let us note that terminating the verifications upon failure of the next checked condition in (2) increases the efficiency of solving the situational problem compared to initially solving all predicate functions sc_i and computing the situational problem as $SP = \bigwedge_1^n sc_i$. Let us note that when solving situational problems, applying Horner's rule as $SP = \bigwedge_1^k sc_i,\ k < n$, instead of checking all conditions as $SP = \bigwedge_1^n sc_i$ improves the efficiency in obtaining the final result. Applying (1) within a thirty-day simulation interval with a 10-sec step will lead to 777,600 and 1,036,800 checks for each of the first two situational problems, respectively.

Table 2 and Table 3 show the performance when applying these two approaches – Eq. (2) and Eq. (4), evaluated for different initial orderings of conditions in the two situational problems. It can be seen that the initial order of conditions affects the efficiency when applying Horner's rule. Applying the proposed heuristic results in nearly equalising the number of checks.

Even with an increase in the number of conditions in situational problems, the number of checks is decreasing. When applying Horner's rule, the number of checks is sensitive to the order of situational conditions and the orbital parameters of the satellite. The value of the orbital parameters of the satellites, as well as the position of the Sun and the Moon, determines the geometric configuration with which the situational conditions are associated. The reordered heuristic minimises this and other irregularities.

Discussion

Solving a situational problem involving several conditions, for an extended interval of simulation time, is not based on an exact algorithm, even when applying Horner's rule. The number of verified conditions for different moments of time t varies. Thus, the algorithmic complexity changes depending on the checked conditions – $O(m(t))$. Only for the time interval when all conditions are fulfilled, the complexity is $O(n(t))$. Applying conditions reordering, the algorithmic complexity is O(1) for long segments along the time axis!

Conclusion and outlook

The proposed approach for efficiency enhancement of the situational analysis is implemented within the framework of the developed PSAS. The latest version of the situational solver is designed for use in a multiphysics environment for space mission simulations. The application of this approach turns the solver into a powerful analysis tool, increasing the efficiency of simulations.

Future development of the parallel situational analysis solver involves developing and including other types of situational conditions and algorithms to enhance its efficiency. Given the formal nature of the presented approach for situational analysis, a search for other application possibilities is also foreseen. For example, work is being done on applying the situational solver to problems related to the self-shadowing of solar panels on structurally complex space platforms.

Acknowledgement

The author would like to express his gratitude to fortune for the opportunity to collaborate, many years ago, with Dr Alexander Sukhanov from the Space Research Institute at the Russian Academy of Sciences, which subsequently led to the idea proposed in this manuscript.

Appendix A

```
 type     SitCond
   integer    sit_code      ! Code of the situation condition type
   integer    sat_num       ! The identification number of a satellite is concerned
                            ! with this situational condition
   logical    begin_sit     ! The situational interval begins
   logical    fl_rezults    ! If .true. - end of the situational interval
   real       duration      ! Duration of a current situational condition/event
   real       dt_sit        ! Accumulates duration of situational condition
 union
  map                       ! Sit__1: Passing over circular region of Earth surface
    integer    reg_num          ! Number of regions- actual number
    real       lati_r , longi_r ! Geographic coordinates of the centre
    real       angle_r          ! Angular radius of the region
  end map
  map                       ! Sit__6: Circular geographic region - day[/night] time
    real       lati_6,longi_6   ! Coordinates of the centre
    real       angle_6          ! Angular radius
  end map
  map                       ! Sit_76,77: The moon is under horizon
    logical    Moon_under       ! Moon's "umbra"
    logical    Moon_horizon     ! Moon on horizon (rise/set)
    logical    Moon_over        ! Moon's light
  end map
  map                       ! Sit_78/79: satellite in conic shadow
    real       num_sat_79
    logical    umbra            ! If .true. - <umbra>
    logical    penumbra         ! If .true. - <penumbra>
    logical    sunligh          ! If .true. - penumbra=.false, umbra=.false.
  end map
 end union
end type  SitCond
```

```
type     sit_problem
  union
    map                          ! For the zero element; only for solving process control
      integer     SP_type          ! Type of situation problem
      integer  max_cond           ! Maximum number of conditions in the problem
      integer    efficiency        ! Efficiency-improving algorithm: 0- none, 1/2/3
      logical    begin_sit
      logical    fl_rezults         ! If is .true. - end of time window
      real*8      t1,t2              ! Time interval where the situation is satisfied
      real        duration,dt_sit  ! Duration of the situational problem time window
      real      t_task_total       ! Accumulates total durations for the simulation period
    end map
    map
      type (SitCond) sit_cond
    end map
  end union
end type sit_problem
```

References

1. Abbasi, V., Thorsteinson, S., Balam, D., Rowe, J., Laurin, D., Scott, L. and Doyon, M., (2019), January. The NEOSSat experience: 5 years in the life of Canada's space surveillance telescope. In *1st NEO and Debris Detection Conference* (Vol. 22).
2. Akhmetov, R., Filatov, A., Khalilov, R., Raube, S., Borisov, M., Salmin, V., Tkachenko, I., Safronov, S. and Ivanushkin, M., (2023), "AIST-2D": Results of flight tests and application of earth remote sensing data for solving thematic problems. *The Egyptian Journal of Remote Sensing and Space Science*, *26* (3), pp. 427–454.
3. Ambrosio, A., Cardoso, P. and Orlando, V., (2006), Brazilian satellite simulators: previous solutions trade-off and new perpectives for the CBERS program. In *SpaceOps 2006 Conference*, p. 5916.
4. ASTOS [Online]. Available at: https://www.astos.de/products/astos
5. Atanassov, A.M., (2008), Enhancing the Efficiency in Checking Constraints Satisfaction when Planning Ground-based and Space Experiments, Using an Alternative Problem, *Aerospace Research in Bulgaria*, 22, pp. 51–58. http://journal.space.bas.bg/arhiv/n%2022/Articles/4_Atanassov.pdf
6. Atanassov, A.M., (2016), Parallel satellite orbital situational problems solver for space missions design and control. *Advances in Space Research*, *58* (9), pp. 1819–1826.
7. Atanassov, A.M., (2024), Developing Situational Conditions and Program Codes for Parallel Situational Analysis Solver based on Conic Umbra/Sunlit Models. *Aerospace Research in Bulgaria*, *36*, pp. 15–28. Available at: *http://journal.space.bas.bg/arhiv/n%2036/Articles/2_Atanassov.pdf*
8. Atanassov, A., AN APPLICATION OF THE SITUATIONAL SOLVER TO PROBLEMS CONTAINING CONDITION RELATED TO VISIBILITY OF THE MOON BY A SATELLITE. Proceedings of SES 2022, Sofia, Bulgaria, pp. 62–67. Available at: http://space.bas.bg/SES/archive/SES%202022_DOKLADI/1_Space%20Physics/11_Atanasov.pdf

9. Bernacca, P.L., Facchinetti, C. and Fantino, E., (2010), Star visibility and tracking from the Space Station. *Advances in space research*, *46* (11), pp. 1354–1381.
10. Chagas, R.A., de Sousa, F.L., Louro, A.C. and dos Santos, W.G., (2019), Modeling and design of a multidisciplinary simulator of the concept of operations for space mission pre-phase A studies. *Concurrent Engineering*, *27* (1), pp. 28–39.
11. Colagrossi, A., Prinetto, J., Silvestrini, S. and Lavagna, M., (2020), Sky visibility analysis for astrophysical data return maximization in HERMES constellation. *Journal of Astronomical Telescopes, Instruments, and Systems*, *6* (4), pp. 048001–048001. doi: 10.1117/1.JATIS.6.4.048001
12. Du, J., Lei, X. and Sang, J., (2019a), A space surveillance satellite for cataloging high-altitude small debris. *Acta Astronautica*, *157*, pp. 268–275.
13. Du, J., Chen, J., Li, B. and Sang, J., (2019b), Tentative design of SBSS constellations for LEO debris catalog maintenance. *Acta Astronautica*, *155*, pp. 379–388.
14. FreeFlyer [Online]. Available at: https://ai-solutions.com/freeflyer-astrodynamic-software/
15. Gaidarov, P., Sukhanov, A., Atanasssov, A., Rjazanova, V., (1989), Planirovanie astrofizicheskih eksperimentov v proekte ”Shipka”, IKI – AN SSSR, Pr.-1534.
16. GMAT [Online]. Available at: https://gmat.atlassian.net/wiki/spaces/GW/overview
17. Kosari, A., Sharifi, A., Ahmadi, A. and Khoshsima, M., (2020), Remote sensing satellite’s attitude control system: rapid performance sizing for passive scan imaging mode. *Aircraft Engineering and Aerospace Technology*, *92* (7), pp. 1073–1083.
18. Lee, B.S., Hwang, Y., Kim, H.Y. and Kim, J., (2006), An Integrated Mission Analysis and Planning System for the KOMPSAT-2 Mission Operations. In *SpaceOps 2006 Conference*, p. 5728.
19. Li, Z., Wang, Y. and Zheng, W., (2020), Space-based optical observations on space debris via multipoint of view. *International Journal of Aerospace Engineering*, *2020*, pp. 1–12.
20. Miller, G., Rosenthal, D., Cohen, W. and Johnston, M., (1987), Expert systems tools for Hubble Space Telescope observation scheduling. *Telematics and Informatics*, *4* (4), pp. 301–311.
21. Miller, G., Johnston, M., Vick, S., Sponsler, J. and Lindenmayer, K., (1988), Knowledge based tools for Hubble Space Telescope planning and scheduling: constraints and strategies. *Telematics and Informatics*, *5* (3), pp. 197–212.
22. Neto, V.V.G., Paes, C.E., Rohling, A.J., Manzano, W. and Nakagawa, E.Y., (2019), April. Modeling & simulation of software architectures of systems-of-systems: An industrial report on the brazilian space system. In *2019 Spring Simulation Conference (SpringSim)*, pp. 1–12, IEEE.
23. Radice, G., Wuilbercq, R. and Cafiero, G., (2014), T-smad: a concurrent design tool for space mission analysis and design. *Frontiers in Aerospace Engineering*, *3* (1), pp. 1–6.
24. Pandiyan, R., Subbarao, S.V., Nagamani, T., Rao, C., Rao, N.H.P., Joglekar, H., Kumar, N., Dumpa, S.R.P., Chauhan, A. and Dakshayani, B.P., (2017), Planning and scheduling of payloads of AstroSat during initial and normal phase observations. *Journal of Astrophysics and Astronomy*, *38*, pp. 1–20.

25. Pemberton, J., Galiber F., (2001), A Constraint-Based Approach to Satellite Scheduling. In E. Freuder and R. Wallace, editors, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science: Constraint Programming and Large Scale Discrete Optimization* 57:101–114.
26. Poghosyan, A., Lluch, I., Matevosyan, H., Lamb, A., Moreno, C. A., Taylor, C., ... & Camps, A., (2016), Unified classification for distributed satellite systems. In *4th International Federated and Fractionated Satellite Systems Workshop, Rome, Italy*.
27. Prech L, Ruzhin YY, Dokukin VS, Nemecek Z and Safrankova J., (2018), Overview of APEX Project Results. Front. Astron. Space Sci. 5:46. doi: 10.3389/fspas.2018.00046
28. Prohorenko V. I., (1985), Situatsionnyy analiz orbit khvostovogo i avroralnogo zondov v proyekte "Interbol", IKI – AN SSSR, Pr.-1037.
29. Rabelo, L., Sala-Diakanda, S., Pastrana, J., Marin, M., Bhide, S., Joledo, O. and Bardina, J., (2013), Simulation modeling of space missions using the high level architecture. *Modelling and Simulation in Engineering*, *2013*, pp. 11–11.
30. Rauber, T., Rünger G., (2010), Parallel Programming: For Multicore and Cluster Systems. Springer.
31. Scott, R, Thorsteinson, S., (2018), Key Findings from the NEOSSat Space-Based SSA Microsatellite Mission. Available on: https://amostech.com/TechnicalPapers/2018/Space-Based_Assets/Scott.pdf
32. Selva, D., A. Golkar, O. Korobova, I. L. I. Cruz, P. Collopy, & O. L. de Weck, (2017), Distributed earth satellite systems: What is needed to move forward?. *Journal of Aerospace Information Systems*, *14* (8), pp. 412–438.
33. Song, Y.J., Lee, D., Bae, J.H., Kim, B.Y., Kim, Y., Lee, E.J., Kim, H.J. and Park, S.Y., (2016), Preliminary design of LUDOLP: the flight dynamics subsystem for the Korea Pathfinder Lunar Orbiter mission. In *14th International Conference on Space Operations*, p. 2311.
34. STK [Online]. Available at: https://www.ansys.com/content/dam/amp/2023/november/asset-creation/stk-capabilities.pdf
35. Wertz, J.R., Larson, W.J., (1999), Space Mission Analysis and Design. Microcosm Press, Kluwer Academic Publishers, third ed.
36. Williams, B.S. and Gage, K.R., (2006), June. The Chandra X-ray Observatory mission planning process: managing and implementing constraints. In *Observatory Operations: Strategies, Processes, and Systems,* Vol. 6270, pp. 726–737, SPIE.
37. ”XMM-Newton Users Handbook”, Issue 2.22, 2024 (ESA: XMM-Newton SOC). Avalaible on: https://xmm-tools.cosmos.esa.int/external/xmm_user_support/documentation/uhb/XMM_UHB.pdf

ПРИМЕНЕНИЕ ПЕРЕСТАВЛЕННОЙ ЭВРИСТИКИ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ СИТУАЦИОННОГО АНАЛИЗА ПРИ РЕШЕНИИ ЗАДАЧ, СОДЕРЖАЩИХ БОЛЕЕ ОДНОГО УСЛОВИЯ, В АНАЛИЗЕ КОСМИЧЕСКИХ МИССИЙ

Атанас Атанасов

Резюме

Ситуационный анализ занимается проверкой условий геометрического или физического характера во время движения спутников по орбитам. Этот анализ применяется для определения временных интервалов, оптимальных для выполнения конкретных операций со спутниками. Ситуационные задачи решаются на различных этапах подготовки и эксплуатации спутниковых миссий. Наиболее интересными являются случаи, когда ситуационная задача содержит несколько условий. Каждая ситуационная задача решается путем последовательной проверки соответствующих условий. Проверки могут быть прекращены при обнаружении невыполненного условия (правило Хорнера) для повышения эффективности анализа. Вычисления повторяются для последующих временных шагов. Решение такой ситуационной задачи может быть трудоемким при проведении многовариантных и многоспутниковых симуляций. Предлагается эвристический подход для повышения эффективности ситуационного анализа задач, включающих несколько условий. Этот подход основан на переупорядочивании проверок условий. Подход алгоритмизирован и интегрирован в ранее разработанный решатель ситуационного анализа. Эффективность предложенной эвристики оценивается на примерах с различным количеством ситуационных условий. Этот подход демонстрирует большую эффективность, чем применение правила Хорнера.