Heuristic Algorithm for Solving the Cosmonauts Training Planning Problem

Alexander A. Lazarev
Institute of Control Sciences
65 Profsoyuznaya street,
117997 Moscow, Russia
jobmath@mail.ru

Nail Khusnullin
Institute of Control Sciences
65 Profsoyuznaya street,
117997 Moscow, Russia
nhusnullin@gmail.com

Elena Musatova
Institute of Control Sciences
65 Profsoyuznaya street,
117997 Moscow, Russia
nekolyap@mail.ru

Denis Yadrentsev
YU.A. Gagarin Research and Test Cosmonaut Training Center
Star City, 141160, Russia,
d.yadrentsev@gctc.ru

Konstantin Ponomarev
YU.A. Gagarin Research and Test Cosmonaut Training Center
Star City, 141160, Russia,
K.Ponomarev@gctc.ru

Abstract

The cosmonauts training planning problem is a problem of construction of cosmonauts training timetable. Each cosmonaut has his own set of tasks which should be performed with respect to resource and time constraints. The problem is to determine start moments for all considered tasks. This problem is a generalization of the resource-constrained project scheduling problem with "time windows". A new heuristic method based on constraint programming is developed. The effectiveness of the method is verified on real data.

1 Introduction

We consider a problem of planning the International Space Station (ISS) cosmonauts training. In order to maintain reliability of a space flight, the crew members are obligated to be trained for different types of situations and operations, to obtain required skills and knowledge before the launch. Hence, the Yu.A. Gagarin Research & Test Cosmonaut Training Center (CTC) must plan and schedule a list of trainings for every cosmonaut. The sequence of the training program is based on the following training phases:

- general space training (GST) of candidates for cosmonauts;
- training in groups, separated by the type of manned spacecraft (MSC) or areas of specialization;
- training in approved crews for a specific space flight on MSC.

Passing the sequence of the stages of the training is mandatory for all Russian cosmonauts. The GST is performed for every candidate only once. The other stages can be performed repeatedly. Usually, the phases last 2, 2 and

In: Yu. G. Evtushenko, M. Yu. Khachay, O. V. Khamisov, Yu. A. Kochetov, V.U. Malkova, M.A. Posypkin (eds.): Proceedings of the OPTIMA-2017 Conference, Petrovac, Montenegro, 02-Oct-2017, published at http://ceur-ws.org

2.5 years, respectively. This article is devoted to the third stage. In general, three crew qualification levels are defined; a user level, an operator level and a specialist level. For a given flight program, for every on-board system, a pre-defined set of minimum qualifications is needed to safely operate and maintain the system (for example, one specialist, one operator and one user). Each crew member, while being a specialist for one system, will be an operator or only a user for another one system. Consequently, the training program for each crew member is individually tailored to his set of tasks and pre-defined qualification levels. The development of training plans for a crew is another problem. Some results for its solving can be found in [Lazarev et al., 2016]. In this article we believe that an individual plan for each cosmonaut already exists, and the problem is to determine start moments for all considered tasks for each cosmonaut.

Previously, for solving this problem we have proposed an approach based on methods of integer linear programming [Musatova et al., 2016]. However, this approach turned out to be ineffective for high-dimensional problems. In [Lazarev et al., 2016] comparison of two approaches for a medium dimension cosmonaut training problem was presented. The first approach was based on integer linear programming and the second one was on the basis of constraint programming (CP) [Dechter, 2003]. It has been shown an obvious advantage of CP. We can explain the benefits of the CP by a large number of different constraints imposed by the training process.

The paper has the following structure. In section 2 a mathematical formulation of the considered problem is given. Section 3 is devoted to properties of the problem. On the basis of these properties in Section 4 a heuristic algorithm for solving the problem is proposed. Results of a computational experiment for different levels of a crew experience are given.

2 Problem Statement

This section provides a formulation of the problem, that arises in cosmonauts training scheduling. Simultaneously there are several crews on training in the Cosmonaut Training Center, but this article presents a model and a method for its solving for one of them. We will assume that the planning of the cosmonauts training takes place at a certain given time interval, which specifies a planning horizon. We take as a unit of time a half-hour interval. Denote as W a set of all weeks and as D a set of all days on the planning horizon. Set H is a set of all half-hour intervals in a day. Possible moments of the beginning of a task are in a set $T = \{0, 1, \dots, \mathcal{T}\}$. Each time moment $t \in T$ is characterized by its number of the week $w(t) \in W$ on the planning horizon, by its number of the day $d(t) \in D$ and by a number of the half-hour interval $h(t)$ in the day $d(t)$.

Let J be a set of all stages of cosmonauts training (set of tasks) and duration of task $j \in J$ is equal to p_j units of time. We input a variable S_j, the value of which is equal to the moment of the beginning of the task j. Let $G = (J, \Gamma)$ be a graph of precedence relations between tasks. If $(i, j) \in \Gamma$, then the task i has to be completed before the beginning of the task j:

$$S_j - S_i \geq p_i \;\; \forall (i, j) \in \Gamma. \tag{1}$$

We input a set of tasks which are active at the time moment t:

$$A_t = \{j \in J \mid S_j \leq t < S_j + p_j\}. \tag{2}$$

Let R be a set of renewable resources (instructors, simulators, classrooms, special equipment). Each cosmonaut is also a resource, available in a quantity of 1 during a working day (excluding holidays, business trips, etc.). Denote by ra_{rt} the amount of resource $r \in R$ available at time t, and by rc_{jr} — the amount of resource $r \in R$ required for the task $j \in J$. Then the resource constraints can be written in the following form:

$$\sum_{j \in A_t} rc_{jr} \leq ra_{rt} \;\; \forall r \in R, \; \forall t \in T. \tag{3}$$

Constraints (1), (3) are the standard constraints of the Resource-Constrained Project Scheduling Problem (RCPSP). As a rule, the objective function in this problem is makespan (project duration). RCPSP is strongly NP-hard [Artigues et al., 2008]. By this reason different heuristic methods are proposed for its solving: priority-rule based scheduling methods, truncated branch-and-bound, integer programming based heuristics, disjunctive arc concepts, local constraint-based analysis, sampling techniques, evolutionary algorithms and local search techniques (see, for example, [Brucker et al., 1999, Kolisch & Hartmann, 2006, Valls et al., 2003, Debels & Vanhoucke, 2007, Homberger, 2007]. In [Kolisch & Hartmann, 2006] a large number of heuristics that have been proposed in the literature for RCPSP solving are summarized and categorized. However, in the cosmonauts training problem there are many other additional constraints that lead to additional difficulties for solving process. In the case of existence of additional constraints, called "time windows" or "minimum and maximum time lags" (which will be described below), the decision problem of a feasible solution existence is strongly NP-complete [Bartusch et al., 1988]. Below we list the additional restrictions that arise in the problem under consideration.

Any task has to be completed before the end of a working day:

$$h(S_j) \leq |H| - p_j + 1 \ \ \forall j \in J. \tag{4}$$

We divide the operations that take more than one day into one-day operations connected with special strict precedence relations described below.

Denote as I a set of all cosmonauts of the crew under scrutiny. Each cosmonaut $i \in I$ has his own set of tasks J^i, $\cup_{i \in I} J^i = J$. For some sets of tasks there are restrictions on total duration of tasks per day, or per week for one cosmonaut:

$$\sum_{j \in A_k^i, d(S_j)=d} p_j \leq a_k \ \ \forall i \in I, \ \forall d \in D, \ \forall k \in \{1, 2, \ldots, K_a^i\}, \tag{5}$$

$$\sum_{j \in B_k^i, w(S_j)=w} p_j \leq b_k \ \ \forall i \in I, \ \forall w \in W, \ \forall k \in \{1, 2, \ldots, K_b^i\}. \tag{6}$$

It means that any cosmonaut i can have no more that a_k time units of tasks from the subset A_k^i per day and no more b_k time units of tasks from the subset B_k^i per week, where K_a^i and K_b^i are numbers of such subsets. Such restrictions may be caused by medical standards (for example, a cosmonaut can not be engaged in physical activity more than a certain number of hours per day) or features of the educational processes.

Each instructor has a workload of no more than c_k time units per day:

$$\sum_{j \in J^R(r), d(S_j)=d} p_j \leq c_k \ \ \forall d \in D, \ \forall r \in R^I, \tag{7}$$

where R^I is a set of instructors and $J^R(r)$ is a set of tasks, for which resource r is required.

For all tasks j from a special set J^t time boundaries $[t_j^1; t_j^2]$ are established:

$$t_j^1 \leq S_j \leq t_j^2 \ \ \forall j \in J^t. \tag{8}$$

These boundaries can be both precise and rather extended. In the first case a task has an exact date of its execution and in the second case — long period of time (for example, a winter forest landing training has to be in winter).

Some tasks can be performed only in some period of a day (for example, exams have to be in the morning):

$$h(S_j) \in H^j \ \ \forall j \in J^d, \tag{9}$$

where J^d is a set of tasks with day restrictions and H^j is a set of possible beginning moments of the task j. Sometimes it is more convenient to write down this restriction as "some tasks cannot be held during certain parts of a day" (for example, physical training can not be performed immediately after lunch):

$$h(S_j) \notin \bar{H}^j \ \ \forall j \in J^d, \tag{10}$$

where $\bar{H}^j$ is a set of impossible beginning moments of the task j.

Some tasks are performed by all crew members simultaneously:

$$S_{j_1} = S_{j_2} = S_{j_3} \ \ \forall (j_1, j_2, j_3) \in J^{123}, \tag{11}$$

where J^{123} is a set of triples of tasks that have to be simultaneously. Similarly, some tasks are conducted at the same time for two crew members:

$$S_{j_1} = S_{j_2} \ \ \forall (j_1, j_2) \in J^{12}. \tag{12}$$

In addition to the graph of precedence relations between tasks a graph of strict precedence relations $G' = (J, \Gamma')$ is introduced. We have $(j_1, j_2) \in \Gamma'$ if task j_2 must be performed strictly after g'_{j_1,j_2} intervals after the task j_1:

$$S_{j_2} = S_{j_1} + p_{j_1} + g'_{j_1,j_2} \ \ \forall (j_1, j_2) \in \Gamma'. \tag{13}$$

In some cases a time distance between two tasks is given in weeks with help of another graph $G'' = (J, \Gamma'')$:

$$w(S_{j_2}) = w(S_{j_1}) + g''_{j_1,j_2} \ \ \forall (j_1, j_2) \in \Gamma''. \tag{14}$$

Every week a cosmonaut should remain f^i time units for administrative duties (for example, self-preparation, work with documentation, etc.):

$$WL - \sum_{j \in J^i, w(S_j)=w} p_j \geq f^i \ \ \forall i \in I, \ \forall w \in W, \tag{15}$$

where WL is a weekly workload for a cosmonaut.

Our problem is to find a feasible solution under constraints (1),(3)–(15). Note that we do not formulate the optimization problem. Schedule of cosmonauts training for a large planning horizon is required for the staff of the CTC for strategic planning. Solving a problem of optimizing a current weekly or monthly schedule (minimizing training costs, maximizing the priorities of instructors or cosmonauts) is a tactical problem and the further direction of our research.

3 Structure of the Input Data

As the planning horizon is very large, a solution of the problem (1),(3)–(15) cannot be obtained directly by modern solvers. However, taking into account already existing schedules and researching the process of scheduling of the Cosmonaut Training Center specialists have allowed us to obtain additional information about the structure of the problem. Analysis of the existing schedules showed the following features.

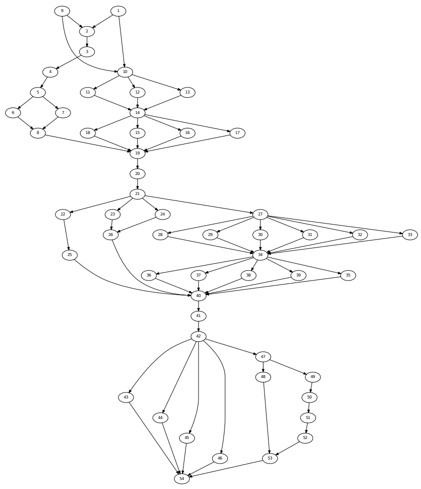

Figure 1: Graph of precedence relations of on-board systems

- Most of the tasks are organized in the so-called on-board systems. An on-board system is a set of tasks united by a theme.
- The graph G of the precedence relations at the macro level is given by a graph of the on-board systems (see figure 1). Each vertex of the graph presented on the picture is, in its turn, also graph of tasks of an on-board system.
- The graph G allows some decomposition, because at some key time points all tasks of some on-board systems must be completed, and the remaining on-board systems must not start yet.
- Specialists of the Cosmonaut Training Center identify priority on-board systems, the schedule for which must be constructed in the first place. These on-board systems are crucial because they require a lot of resources.
- There is a some set of periodical tasks that are not connected by precedence relations and have to be in the schedule every week. As a rule, these tasks do not require special equipment and resources which are used for the study of on-board systems, but they require restrictions like (5), (6), (9).

4 Numerical Results

Taking into account the above observations, we propose to build a feasible schedule as follows. Denote as J_{per} a set of all tasks that are isolated vertices of the graph G. We divide the set $J \backslash J_{per}$ into subsets J_1, $J_2 \ldots$, J_{m-1}, J_m. For any J_k and J_l, where $k < l$, either all on-board systems from J_k are more prioritized than on-board systems from J_l, or all tasks from the set J_k must be completed earlier than any of the tasks of the set J_l.

Let S be a set of start time moments of tasks from some set J_k. Function $Schedule(S, J_k, J_l, J_u)$ returns time moments of tasks of J_l from a certain schedule σ of $J_k \cup J_l \cup J_u$ subject to the following conditions:

- start moment $S_j(\sigma)$ of any task $j \in J_k$ is fixed, $S_j(\sigma) \in S$;
- all constraints for tasks from the set $J_k \cup J_l \cup J_u$ are fulfilled.

In other words, we (1) fix start moments of J_k, (2) build a schedule for $J_l \cup J_u$ provided that all conditions for $J_k \cup J_l \cup J_u$ are fulfilled and (3) save time moments for J_l. If there is no such schedule, the function returns an appropriate message. To calculate this function, we used solver IBM ILOG CPLEX CP 12.6.2[1]. Below we present an algorithm for constructing a feasible solution of problem (1),(3)–(15).

Algorithm 1 Feasible solution building

Precondition: $\bar{J} = \{J_1, J_2, \ldots, J_m\}$, J_{per}

function RUN($\bar{J}$, J_{per})
    $S \leftarrow Schedule(\emptyset, \emptyset, J^t, J_{per})$
    $J_{fix} \leftarrow J^t$
    for $i \leftarrow 1$ to m **do**
        $S \leftarrow S \cup Schedule(S, J_{fix}, J_i, J_{per})$
        $J_{fix} \leftarrow J_{fix} \cup J_i$
    end for
    return $S \cup Schedule(S, J_{fix}, J_{per}, \emptyset)$
end function

Thus, at each step of the algorithm we add to the already existing schedule a new set of on-board systems that either need to be studied later than the already considered ones, or can be studied in parallel with them, but have a lower priority. In addition, at each iteration, periodical tasks are added, but the moments of their starts are not fixed for the next iteration.

In table 1 numerical results are presented. We tested our algorithm on two problems with real world input data. The first problem is a cosmonauts training planning problem for the case when all cosmonauts are experienced. It means that each cosmonaut in the crew has the minimum possible set of tasks. The second problem is for all inexperienced crew members. It is a case of maximum possible size of the set J. We use the following notations: N is a number of the problem, *Plan.horizon* is a number of weeks in the schedule, *Var.* is a number of variables in the problem, *Constr.* is a number of constraints, *Branches* is a number of inner branches of the solver, *Time* is a runtime of the algorithm. Number m of subsets J_i was equaled to 10. The calculations were performed on a workstation with an Intel Xeon processor E5-2673, 2.4GHz and 15Gb of RAM.

Table 1: Numerical results

N	Plan.horizon	Var.	Const.	Branches	Time, m:c
1	21	4572	149997	123613	3:45.46
2	60	6963	156168	392026	9:56.16

5 Conclusion

The article presents a mathematical formulation of the cosmonauts training planning problem. An algorithm based on CP for finding a feasible solution of the problem is proposed. The algorithm was tested on real data

[1]http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

for a large planning horizon. However, we did not consider a complete set of on-board systems, since we do not yet have real data for the remaining cosmonauts trainings. The solving of the cosmonauts training planning problem on the entire horizon of planning is our priority goal. Also our further research will be devoted to the construction of optimal weekly and monthly schedules with a known set of tasks.

Acknowledgements

This work was supported by the Russian Science Foundation (grant 17-19-01665).

References

[Musatova et al., 2016] Musatova, E., Lazarev A., Ponomarev K., Yadrentsev D., Bronnikov S., & Khusnullin N. (2016). A Mathematical Model for the Astronaut Training Scheduling Problem. *IFAC-PapersOnLine, 49(12)*, 221-225.

[Bronnikov et al., 2015] Bronnikov, S., Dolgui, A., Lazarev, A., Morozov N., Petrov A., Sadykov R., Sologub A., Werner F., Yadrentsev D., Musatova E., & Khusnullin N. (2015). *Approaches for Planning the ISS Cosmonaut Training.* Preprint 12/15, Faculty of mathematics, Otto-von-Guericke-University Magdeburg.

[Lazarev et al., 2016] Lazarev, A.A., Bronnikov, S.V., Gerasimov, A.R., Musatova, E.G., Petrov, A.S., Ponomarev K.V., Kharlamov, M., Khusnullin, N.F., & Yadrentsev. D.A. (2016). Mathematical modeling of the astronaut training scheduling. *UBS, 63*, 129154.

[Dechter, 2003] Dechter, R. (2003). *Constraint Processing.* San Francisco, USA: Morgan Kaufmann Publishers.

[Artigues et al., 2008] Ed. Artigues, C., Demassey, S., & Neron, E. (2008). *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications.* Wiley-ISTE.

[Bartusch et al., 1988] Bartusch, M., Mohring, R.H., & Radermache, F.J. (1988). Scheduling project networks with resource constraints and time windows *Annals of Operations Research, 16*, 201-240.

[Brucker et al., 1999] Brucker, P., Drexl, A., Mohring, R., Neumann, K., Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research, 112*, 3-41.

[Kolisch & Hartmann, 2006] Kolisch, R, & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research, 174(1)*, 23-37.

[Valls et al., 2003] Valls, V., Quintanilla, M.S., Ballestin, F. (2003). Resource-constrained project scheduling: A critical activity reordering heuristic. *European Journal of Operational Research, 149(2)*, 282-301.

[Homberger, 2007] Homberger, J. (2007). A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. *Intl. Trans. in Op. Res., 14*, 565-589.

[Debels & Vanhoucke, 2007] Debels, D., & Vanhoucke, M. (2007). A Decomposition-Based Genetic Algorithm for the Resource-Constrained Project-Scheduling Problem. *Operations Research, 55(3)*, 457-469.