Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-12 (2016) 221–225

A Mathematical Model for the Astronaut Training Scheduling Problem

Elena G. Musatova [*] Alexander A. Lazarev [**]
Konstantin V. Ponomarev [***] Denis A. Yadrentsev [****]
Sergey V. Bronnikov [†] Nail F. Khusnullin [‡]

[*] *V.A. Trapeznikov Institute of Control Science of Russian Academy of Sciences, Moscow, Russia (e-mail: nekolyap@ mail.ru)*
[**] *V.A. Trapeznikov Institute of Control Science of Russian Academy of Sciences; Lomonosov Moscow State University; Moscow Institute of Physics and Technology; International Laboratory of Decision Choise and Analysis, National Research University Higher School of Economics, Moscow, Russia (e-mail: jobmath@mail.ru)*
[***] *YU. A. Gagarin Research & Test Cosmonaut Training Center, Star City, Russia (e-mail: k.ponomarev@gctc.ru)*
[****] *YU. A. Gagarin Research & Test Cosmonaut Training Center, Star City, Russia (e-mail: D.Yadrentsev@gctc.ru)*
[†] *Rocket and Space Corporation Energia after S.P. Korolev, Korolev, Russia (e-mail: sbronnik@mail.ru)*
[‡] *V.A. Trapeznikov Institute of Control Science of Russian Academy of Sciences, Moscow, Russia (e-mail: nhusnullin@gmail.com)*

Abstract: We consider a problem of the astronaut training scheduling. Each astronaut 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. For this issue a mathematical model based on integer linear programming is proposed. Computational results of the implemented model and experiments on real data are presented.

Keywords: Operations Research, scheduling models, integer linear programming.

1. INTRODUCTION

There is a wide range of tough issues in modern astronautics. All these issues require careful planning of all activities in order to avoid organizational mistakes. What is more important, all activities have to be performed before the given deadlines. One of such issues is the problem of preparing astronaut crews for working on the International Space Station (ISS). The astronaut training is a very long, expensive and complex process. A spacecraft is one of the most complex peaces of equipment which has been ever built. Safety and security are the crucial points of any spaceflight. Therefore, all astronauts need to have strong background in many spheres. So, the main purpose of the astronauts training is to achieve necessary skills and knowledge.

Nowadays, in Russia, spaceflight training scheduling is performed manually and without using any mathematical approaches. Due to that, fast changes of a training plan will cause a huge workload. We hope that the considered approaches and models will lead to reducing these workloads.

Commonly, the astronaut training planning is divided into the two stages: the volume planning and the timetabling. In the former one, for each astronaut a set of tasks is formed depending on requirements of their qualifications and forthcoming on-board experiments complexity conditions. For details, see at Bronnikov et al. (2015a), Bronnikov et al. (2015b).

At the stage of timetabling, for each astronaut the schedule should be compiled: for each task the start time of its performance should be determined, see Bronnikov et al. (2015c). It is assumed that for each astronaut the set of tasks is founded at the previous stage.

From a mathematical point of view, the spaceflight training scheduling can be considered as a generalization of the resource-constrained project scheduling problem, see Artigues (2008). This problem is NP-hard. In practice, a planning horizon is about 3 years. Each astronaut has an individual, aperiodical learning plan. So, the problem has a very large dimension and is hard to solve. In this paper a mathematical model based on integer linear programming (see Wolsey et al. (1988)) is proposed. Computational results is the best evidence of the applied approach.

2. PROBLEM STATEMENT

There is a set of crews. Each crew consists of a number of astronauts. Each astronaut has his own set of training tasks. Dates of the training start and finish are given. The goal is to form a training schedule for each astronaut.

There are the following constraints:

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.601

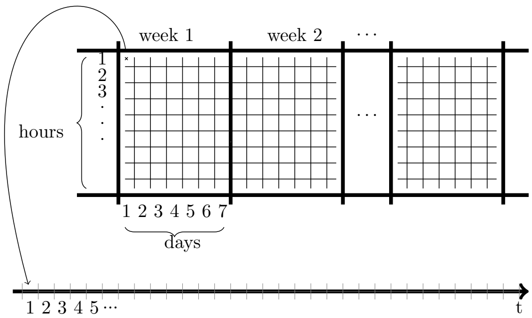


Fig. 1. Time intervals

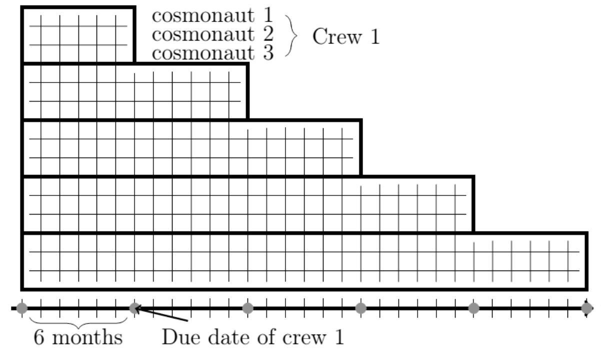


Fig. 2. Training schedule for all crews.

- each astronaut should perform all required tasks;
- physiological aspects of training should be properly taken into account;
- the partial order of tasks is given;
- training resources (teachers, simulators and etc) are restricted;
- some tasks have fixed start times.

2.1 Notations

The following time intervals are introduced:

- W — set of planning weeks, where $|W| = 156$ weeks (3 years);
- $D_w = \{1, 2, 3, 4, 5\}$ — set of work days per week, $w \in W$. The set may be decreased to meet real life requirements (e.g., holidays, day offs, etc.);
- $H_{wd} = \{1, \dots, 18\}$ — set of half-hour intervals of day $d \in D_w$ of week $w \in W$.

It is assumed that the first interval begins at 9.00 a.m. and the latest one ends at 6.00 p.m. In some cases the duration of a work day may be increased if any feasible schedule with the current set of intervals cannot be formed. In this case, schedulers may extend a work day (for instance, for 1 hour) and re-form the schedule in order to provide a solution.

The set of time intervals is designed to work with constraints like "no more than 2 times a week", "in the morning", etc. However, in order to work with constraints linked with task durations let us arrange all triples (w, d, h) in the lexicographical order.

Let us associate each triple to its number: $(w, d, h) \rightarrow t(w, d, h)$ (see Fig. 1):

$$t(w, d, h) = \sum_{w'=1}^{w-1} \sum_{d' \in D_{w'}} |H_{w'd'}| + \sum_{d'=1}^{d-1} |H_{wd'}| + h. \quad (1)$$

We denote the set of all triples (w, d, h) by Y:

$$Y = \{(w, d, h) | w \in W, d \in D_w, h \in H_{wd}\}.$$

The crews start their trainings at different moments (see. Fig. 5). Therefore, over the period of 2.5–3 years, some astronauts have already mastered a part of the operations and thus, each astronaut has his own set of current operations.

Next, the basic notations are introduced.

- C — set of crews.
- K_c — set of astronauts of crew $c \in C$. Usually, $|K_c| = 3$.
- K — set of all astronauts.
- J — set of all tasks.
- J^c — set of tasks of crew $c \in C$.
- J_k — set of tasks of astronaut k, which are required for the implementation of the training plan. We divide this set into the following subsets:
  - J_k^T — set of technical tasks of astronaut k.
  - J_k^F — set of physical training tasks of astronaut k (each task lasts 2 hours or 4 intervals).
  - J_k^A — set of tasks of astronaut k directed to solving administrative issues (self-study, work with documentation).
  - J_k^L — set of language lessons of astronaut k (each task lasts 2 hours or 4 intervals).
- p_j — execution time of task $j \in J$.
- R — set of resources. The set of all astronauts is a subset of R.
- rc_{jr} — amount of resource r needed to perform task j.
- ra_{rwdh} — amount of resource r accessible during time interval h of day d, of week w. Each astronaut is available in amount of 1 at any time.
- e_j, l_j — the earliest and the latest moments at which task $j \in J$ can be performed.
- J_k^{bound} — set of tasks for which time constraints are defined. Due dates can also be described using these boundaries.
- $G = (J, \Gamma)$ – the graph of precedence relations between the tasks. We have $(j, j') \in \Gamma$ if task j must be performed before task j'. With the help of this general graph G individual precedence graphs for each astronaut can be built: $G_k = (J_k, \Gamma_k), k \in K$.
- $H = (J, \mathcal{H})$ – the weighted graph of the strict precedence relations between the tasks. We have $(j, j') \in \mathcal{H}$ if task j' must be performed strictly after $h_{j,j'}$ intervals after the task j. With the help of the graph H individual graphs of the strict precedence relations can be built: $H_k = (J_k, \mathcal{H}_k)$, $k \in K$.

Some tasks are grouped into the butches by studied subjects. In practise, these butches are named on-board systems. Let m_k be the number of on-board systems studied by the astronaut k. So,

$$J_k^{B_1}, J_k^{B_2}, \dots, J_k^{B_{m_k}}$$

are sets of on-board systems studied by the astronaut k.

Let's denote by $Y(k, j)$ the set of all possible time intervals for performing task j by astronaut k. In this case, we do not consider the days when astronaut k is on vacation and consider time constraints (limits e_j and l_j).

We can divide the tasks that take more than one day into one-day tasks. So, we get a sequence of one-day tasks with strict precedence relations.

2.2 Variables

x_{kjwdh} — Boolean variable. We have $x_{kjwdh} = 1$ iff the astronaut k starts the task j from the interval h of the day d of week w.

2.3 Constraints

We agreed that to solve the modeling task, we have established the following constraints.

The resource limits have to be respected.

$$\sum_{k\in K}\sum_{j\in J} rc_{jr} \sum_{\substack{h'>0,\\ h-p_j+1\le h'\le h}} x_{kjwdh'} \le ra_{rwdh}, \tag{2}$$

$$\forall r \in R,\ \forall (w, d, h) \in Y.$$

In this inequality we consider only the operations that are performed at $(w, d, h) \in Y$, i.e., which started in the interval $[h - p_j + 1; h]$ of week w and day d.

Each astronaut should perform all required tasks.

$$\sum_{(w,d,h)\in Y(k,j)} x_{kjwdh} = 1,\ \ \forall k \in K,\ \forall j \in J_k. \tag{3}$$

Each astronaut may have no more than 2 physical trainings per week.

$$\sum_{j\in J_k^F}\sum_{d\in D_w}\sum_{h\in H_{wd}} x_{kjwdh} \le 2,\ \ \forall k \in K, \forall w \in W. \tag{4}$$

Each astronaut may have no more than 2 language lessons per week.

$$\sum_{j\in J_k^L}\sum_{d\in D_w}\sum_{h\in H_{wd}} x_{kjwdh} \le 2,\ \ \forall k \in K, \forall w \in W. \tag{5}$$

Similarly, we have constraints on the administrative issues.

$$\sum_{j\in J_k^A}\sum_{d\in D_w}\sum_{h\in H_{wd}} x_{kjwdh} \le 4,\ \ \forall k \in K, \forall w \in W. \tag{6}$$

It is forbidden to plan more than 4 hours of training for each on-board system per day.

$$\sum_{j\in J_k^{B_i}}\sum_{h\in H_{wd}} p_j x_{kjwdh} \le 8, \tag{7}$$

$$\forall k \in K, \forall i \in \{1, \dots, m_k\}, \forall w \in W, \forall d \in D_w.$$

There are time limits for some tasks.

$$x_{kjwhd} = 0, \tag{8}$$

$$\forall k \in K, \forall j \in J_k^{bound}, \forall (w, d, h) \in Y :\ t(w, d, h) \le e_j - 1,$$

$$x_{kjwhd} = 0, \tag{9}$$

$$\forall k \in K, \forall j \in J_k^{bound}, \forall (w, d, h) \in Y :\ t(w, d, h) \ge l_j + 1.$$

The precedence relations have to be taken into account.

$$\sum_{(w,d,h)\in Y} t(w, d, h)(x_{kj_2wdh} - x_{kj_1wdh}) \ge p_{j_1}, \tag{10}$$

$$\forall k \in K,\ \ \forall (j_1, j_2) \in \Gamma_k.$$

In the same manner strict precedence relations are introduced.

$$\sum_{(w,d,h)\in Y} t(w, d, h)(x_{kj_2wdh} - x_{kj_1wdh}) = p_{j_1} + h_{j_1j_2}, \tag{11}$$

$$\forall k \in K,\ \ \forall (j_1, j_2) \in \mathcal{H}_k.$$

Excluding some time intervals for certain tasks during a day may be applicable in some cases. From this point of view let $[h_1; h_2]$ be the time period of a day when performing some tasks is forbidden. $J_{[h_1;h_2]}$ is a set of such activities. Therefore,

$$\sum_{j\in J_{[h_1;h_2]}}\ \sum_{h_1-p_j+1\le h\le h_2} x_{kjwdh} = 0, \tag{12}$$

$$\forall k \in K,\ \ \forall w \in W,\ \ \forall d \in D_w.$$

For example, if $[h_1; h_2]$ is a lunch time, then $J_{[h_1;h_2]}$ is a set of all tasks, except long tasks which should not be interrupted by lunch.

Another option is the following: $[h_1; h_2]$ is a time period, including lunch time, two hours before and two hours after lunch, $J_{[h_1;h_2]}$ is a set of physical trainings which can not be performed during this interval.

The introducing of a personal time interval $[h_1^k; h_2^k]$ and an individual set of tasks $J_{[h_1^k;h_2^k]}$ is possible for each astronaut.

2.4 Objective function

Our objective is to minimize total training time for each crew.

An additional variable t^{f_c} and constraints on the additional last task $j_k^{f_c}$ for each astronaut $k \in K_c$, $c \in C$ are introduced:

$$t(w, d, h)x_{kj_k^{f_c}wdh} \le t^{f_c},\ \ \forall c \in C, \tag{13}$$

$$\forall k \in K_c,\ \forall (w, d, h) \in Y.$$

The problem is

$$\min \sum_{c\in C} t^{f_c}$$

with regards to (2)–(13).

Since the first crew starts before the others or there are some other reasons, the priority in the planning may be introduced.

Therefore, the weight coefficients may help to formulate the problem like the following:

$$\min \sum_{c\in C} \alpha_c t^{f_c},$$

where α_c is a priority coefficient.

3. IMPLEMENTATION AND COMPUTATIONAL RESULTS

The calculations were performed using the solver IBM ILOG CPLEX. Tab. 1 contains the internal solver information about the initial experiment on small test data. Here $|W|$ is a number of weeks in planning horizon.

Table 1. Problem's parameters

$\|W\|$	Number of constraints	Number of variables	Iterations
1	1724	2859	867
2	3447	10468	1061
3	5238	29952	12982

As we can see from Tab. 1, even for the small planing horizon the problem is characterised by a large dimension.

In order to reduce the complexity of the considered problem, the divide and conquer approach is implemented. The idea is the following: all tasks are grouped into the sessions each of them may be calculated separately. So, a session is a set of tasks which has to be performed by a crew during a given time interval (a few weeks). All sessions are mutually disjoint sets. The rules of partitioning the set of all tasks into the subsets called sessions came from practice and it is not the point of considered problem.

Tab. 2 contains results of the experiment based on real data. Whole planing horizon (18 weeks) is splitted into the 6 sessions, each of them is calculated one by one and merged into one schedule later. The table uses the following notations: column *Session* contains names of sessions, *Time* — computing time in minutes and seconds, $|W|$ — a number of the considered weeks, $|J|$ — number of tasks, *Var.* — a number of variables, *Constr.* — a number of constraints, *Iter.* — a number of iterations.

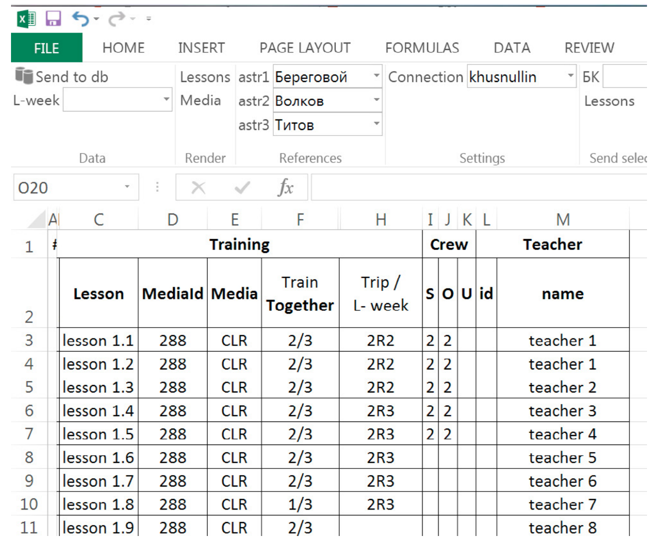
Note that because of the known time limits for the sessions the objective function is not used. Therefore the first found feasible solution is chosen.

Table 2. Schedule for 6 sessions, 18 weeks

Session	Time	$\|W\|$	$\|J\|$	Var.	Constr.	Iter.
R1	0:08.00	3	141	38160	42557	2043
R2	1:51.00	5	237	106815	114807	838186
R3	0:37.00	5	211	95115	110282	232482
R4	3:56.00	5	235	105915	110801	2689598
R5	0:00.30	1	27	2463	5028	0
R6	0:00.60	1	33	3003	5346	0
Total:	7 min	19	884	351471	388823	3762309

At the current moment the spaceflight training management system is developing. There are several significant parts of the system which should be considered thoroughly. Before calculating a schedule, a training load should be formed. For this purpose the plugin for Excel was developed using C# language. Our Excel plugin analyzes the training load data and prepares them for the math solver (Fig.3). Particularly, a planner (the person who forms training load) may have a chance to set resources, time and precedence constraints.

On the next step, a math subsystem based on CPLEX implementation tries to solve the model using training load



#	Lesson	MediaId	Media	Train Together	Trip / L- week	S	O	U	id	name
3	lesson 1.1	288	CLR	2/3	2R2	2	2			teacher 1
4	lesson 1.2	288	CLR	2/3	2R2	2	2			teacher 1
5	lesson 1.3	288	CLR	2/3	2R2	2	2			teacher 2
6	lesson 1.4	288	CLR	2/3	2R3	2	2			teacher 3
7	lesson 1.5	288	CLR	2/3	2R3	2	2			teacher 4
8	lesson 1.6	288	CLR	2/3	2R3					teacher 5
9	lesson 1.7	288	CLR	2/3	2R3					teacher 6
10	lesson 1.8	288	CLR	1/3	2R3					teacher 7
11	lesson 1.9	288	CLR	2/3						teacher 8

Fig. 3. Training load (source data)

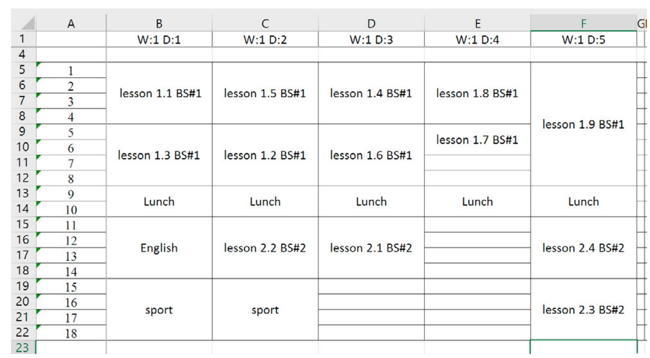


Fig. 4. Time table

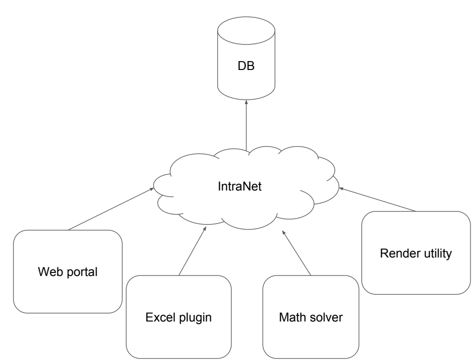


Fig. 5. Solution architecture

data. After that, the CPLEX result matrix is transformed into the human readable format (Fig. 4).

Otherwise a database helps to all modules to exchange data among them. This needs to construct a flexible solution with independent subsystems. In the long term, it may help to replace easily math solver or, for instance, render utility. To summarise everything that was mentioned above, the whole architecture is presented on Fig. 5.

Almost all used components are open source. For instance, database is Postgres, operation system is Ubuntu. However, at the current moment, IBM ILOG CPLEX 12.6.2 is used as a math solver. User may change some settings such as a precedence graph among tasks or planning horizon on the web portal. The web portal is based on python and nginx.

4. FURTHER RESEARCH

The actual problem is the formulation preprocessing to reduce the number of variables x_{kjwdh}. A preprocessing algorithm should eliminate variables x_{kjwdh} that are guaranteed not to participate in any optimal solution. Non-specialized preprocessing algorithms are already built into modern solvers, but specialized algorithms can be much more effective.

An alternative approach to the problem is to use Constraint Programming (CP) solvers. To use this approach, the problem needs to be reformulated as a Constraint Satisfaction Problem (see Dechter (2003)). This can be done quite simply. Instead of Boolean variables x_{kjwdh} we should input variables s_{kj} — the start moment of the task j by astronaut k. The advantage of CP is its possibility to reduce the set of admissible values of variables s_{kj}. The important principle of CP consists of distinguishing constraint propagation and decision-making search. Constraint propagation is a deductive activity which consists in deducing new constraints from existing constraints. The large number of constraints in our problem contributes to high efficiency of CP methods.

Since, in practice, this problem has a very high dimension, we also plan to develop approximate methods for solving this problem.

REFERENCES

Ed. by C. Artigues, S. Demassey, E. Neron. Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications, 2008.

S. Bronnikov, A. Lazarev, N. Morozov, M. Kharlamov and D. Yadrentsev. Mathematical models and approaches in problem of volume planning of ISS astronauts trainings. *28th Conference of the European Chapter on Combinatorial Optimization*, pages 61–62, 2015.

S. Bronnikov, V. Gushchina, A. Lazarev, N. Morozov, A. Sologub and D. Yadrentsev. Three approaches to solving the problem of cosmonauts training planning. *7th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2015)*, pages 750–754, 2015.

S. Bronnikov, A. Dolgui, A. Lazarev, N. Morozov, A. Petrov, R. Sadykov, A. Sologub, F. Werner, D. Yadrentsev, E. Musatova, N. Khusnullin. Approaches for planning the ISS cosmonaut Training. *Preprint 12/15, Faculty of mathematics, Otto-von-Guericke-University Magdeburg*, 2015.

R. Dechter. Constraint Processing. *Morgan Kaufmann*, 2003.

L.A. Wolsey, G.L. Nemhauser. Integer and Combinatorial Optimization. *N.Y.: John Wiley & Sons Inc.*, 1988.