MANAGING A COMPLEX OF JOBS WITH UNCERTAIN EXECUTION REQUEST ARRIVALS

D. A. Kononov* and M. G. Furugyan**

*Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
**Federal Research Center "Computer Science and Control," Russian Academy of Sciences, Moscow, Russia

*✉ dmitrykon52@gmail.com, **✉ rtsccas@yandex.ru

Abstract. This paper considers the scheduling problem for a complex of basic jobs under the condition that at some uncertain times, execution requests for supplementary higher-priority jobs are received. If a supplementary job request arrives during the execution of a basic job, then the latter is terminated and must be restarted at some time upon the complete service of the former. All jobs (basic and supplementary) are executed without interruption. By assumption, during the execution of a basic job, two or more supplementary job requests are unlikely to arrive, and such cases are not analyzed. Also, a supplementary job request can arrive only after the complete service of the previous supplementary request. Two problem formulations are studied as follows. In the first, the performance criterion is the completion time of the basic job complex, and the problem is to minimize this time. In the second formulation, the probability of a collision is minimized, as a situation where a supplementary job request arrives during the execution of a basic job. The problems are solved via their reduction to infinite zero-sum two-player (antagonistic) games and the discrete approximation of the latter by finite games. Model examples are considered. The problem formulation with non-fixed durations of the basic jobs, linearly dependent on the amount of additional resources allocated, is investigated as well. In this case, job scheduling is reduced to a linear programming problem.

Keywords: job scheduling, collision, zero-sum two-player game, antagonistic game, non-renewable resources, mixed strategy, optimal strategy.

INTRODUCTION

Special automated control systems (ACSs) of various classes and purposes are widely used in the development, operation, and management of complex objects. Such systems are oriented to perform several control functions in given fields and have essential properties (system parameters) and characteristics reflecting the efficiency of their realization [1].

In modern conditions, it is important to develop control actions in real-time automated systems. The main characteristic of such problems is an essential upper bound on the time to process input information and output the result, in the form of control actions on the object or messages to the user. The problem becomes even more complicated under uncertainty, when it is necessary to find a schedule in a changing environment, i.e., when new (unpredicted) requests for control signals arrive.

Besides pure science, the above class of problems is of practical value. The need for fast algorithms composing multiprocessor schedules often arises in real-time distributed computing and operational control based on the processing and analysis of incoming real-time data. The following examples can serve as an illustration of the wide practical spread of real-time multiprocessor ACSs and the importance of efficient computation and control algorithms:

• Modern space monitoring systems are real-time systems that continuously process incoming data on the motion of objects in near-Earth space.

• Nuclear reactor control systems at nuclear power plants receive real-time data from many sensors and must promptly implement control actions on the reactor based on these data.

• In developed countries, real-time systems are used in government analytical centers to monitor and analyze continuous economic or environmental information coming from various points. Such systems must efficiently process huge amounts of data and, based on these data, promptly notify of any problems identified in the early stages of occurrence.

• When testing aircraft and other complex technical objects, it is crucial to receive and promptly process, by a real-time computing system, periodic input information about the state of various object nodes.

• At a modern airport with many runways, decision-making includes the assignment of aircraft to different runways as well as their takeoff and landing order.

• In complex logistics systems, decisions must be made in real time when critical situations occur.

• In emergencies, it is necessary to process incoming information urgently and calculate elimination forces and means (make schedules) in real time.

Note that the correctness of a real-time ACS depends not only on the computational accuracy but also on the time to obtain the results. Real-time scheduling is an important part of such systems: the system designer must ensure that all jobs will be executed in due time.

Along with the problem of constructing a feasible schedule for a known real-time computing system, the inverse problem is also topical: design a real-time system of some minimal possible configuration in which a feasible schedule can always be found for a given complex of jobs. This problem is crucial for onboard computing systems, which usually designed by minimizing the necessary computational resources in order to save their mass and power consumption. An algorithm for designing such systems was described in the paper [2].

Job scheduling arises in many spheres of human activity, such as construction, economics, warfare, ecology, mining, management of complex technical objects (airplanes, power plants, and nuclear reactors), transport scheduling, management of computational processes, particularly in real-time systems, and other industries. This topic was widely addressed in the literature. For example, we mention the fundamental monographs [3, 4], where various problems of scheduling theory and discrete optimization were studied, classified (into polynomially solvable and NP-hard), and solved via algorithms. In addition, the authors analyzed the computational complexity of the proposed algorithms. Based on the concept of the distance between problem instances, methods for solving several NP-hard problems with the maximum delay minimization criterion and time-optimal scheduling problems were developed in the book [5].

Various mathematical apparatus is used in scheduling problems. For example, a technique for managing computational processes with directive deadlines was described in the papers [6, 7]. The technique involves finite automata with a stopwatch and time diagrams. This approach is especially relevant in the design of real-time systems.

Job scheduling under uncertainty and risk is of great interest. For example, such problems with non-fixed parameters were investigated in [8–10]. By assumption, the durations of jobs, as well as available and required resources, are given by probabilistic characteristics or their possible ranges. In the latter case, an algorithm for partitioning the set of all possible parameter values into the so-called stability polyhedra was developed. For all parameter values belonging to each such polyhedron, the structure of the optimal schedule remains unchanged. Hence, it is possible to construct a schedule for each polyhedron in advance and choose the necessary solution in real-time computations as soon as the values of uncertain parameters become known. This approach is especially topical in the design and operation of real-time systems with strictly limited computation time.

According to the production planning methodology proposed in [11], the schedule of job execution is compiled together with the analysis of possible changes in production capacities. The original problem was reduced to a nonlinear integer mathematical programming problem. The scheduling problem of job completion dates with the stochastically varying amounts of resources required for job execution was investigated in the paper [12].

This work continues the research initiated in [13, 14]. The scheduling problem of a complex of basic jobs under uncertainty is considered. By assumption, at some uncertain times, there arrive execution requests for supplementary higher-priority jobs. Two problem formulations are studied using a game-theoretic approach. The first one is to minimize the completion time of the basic job complex. In the second formulation, the performance criterion is the probability of no collision. (A collision is a situation where a supplementary job request arrives during the execution of a basic job.) We also investigate the formulation with non-fixed durations of basic jobs, linearly depending on the amount of supplementary resources allocated for this purpose. In this case, a feasible schedule is found by solving a linear programming problem.

1. PROBLEM STATEMENT

There is a complex of basic jobs $W=\{w_1, w_2,\ldots, w_n\}$ with known durations $t_1, t_2,\ldots, t_n$ and a given execution sequence $w_1 \to w_2 \to \ldots \to w_n$.

At some uncertain times $0 \le y_1 \le y_2 \le \ldots \le y_m \le T$, there may arrive execution requests for supplementary jobs $Z = \{z_1, z_2, \ldots, z_m\}$ with known durations $\tau_1, \tau_2, \ldots, \tau_m$, respectively. The upper threshold (deadline)

$$T \ge \sum_{i=1}^{n} t_i.$$

is given.

Supplementary jobs have a higher priority than the basic ones. If a supplementary job request arrives during the execution of a basic job, then the latter is terminated and must be restarted at some time upon the complete service of the former. This situation is called a collision.

All jobs (basic and supplementary) are executed without interruption. By assumption, during the execution of a basic job, two or more supplementary job requests are unlikely to arrive, and such cases are not analyzed. Also, a supplementary job request can arrive only after the complete service of the previous supplementary request. Let the basic and supplementary jobs represent program modules solving some application tasks by an available computing device. It is a renewable resource, i.e., can be reused. When a supplementary job request arrives, this device is immediately passed for its execution. By assumption, in case of no collisions, the basic job must be completed no later than the deadline *T*. We consider two problem formulations as follows.

Problem 1. It is required to design an optimal execution strategy for the basic job complex by minimizing the completion time of the last job w_n (the entire set of jobs, including the supplementary ones).

Problem 2. Assume that the execution requests for the complex of basic jobs *W* and supplementary jobs *Z* are received repeatedly. It is required to design an optimal scheduling strategy for the basic jobs by maximizing the probability of no collision.

Note that in both formulations, the arrival of a request for supplementary job z_j becomes known only at a time y_j, $j = \overline{1, m}$. Such problems arise, e.g., during flight tests. In normal mode, computations are performed using application modules w_i, $i = \overline{1, n}$. At uncertain times y_j, $j = \overline{1, m}$, an abnormal situation may occur, e.g., the values of some important parameters may go beyond admissible limits. In this case, the computations planned are interrupted, and supplementary higher-priority jobs are executed.

We also investigate the problem formulation with non-fixed durations of basic jobs, linearly dependent on the amount of additional (non-renewable) resources allocated.

2. SOLUTION OF PROBLEM 1

First, consider the case $m = 1$. Let a request for supplementary job *z* arrive at a time *y*, and let its duration be τ. We introduce a zero-sum two-player (antagonistic) game with a payoff function $F(x_1, x_2, \ldots, x_n, y)$. In this game, the strategy of the first player determines the times x_i to start the basic jobs $w_i \in W$, $i = \overline{1, n}$, and the strategy of the second player determines the arrival time *y* of the request for the supplementary job *z*. The payoff function is defined as follows:

$$F(x_1, x_2, ..., x_n, y) = \begin{cases} x_n + t_n \text{ if } 0 \le y \le x_1 - \tau, \\ \text{or } x_k + t_k \le y \le x_{k+1} - \tau \\ \text{for some } 1 \le k \le n-1, \\ \text{or } x_n + t_n \le y \le T; \\ \\ y + \tau + \sum_{i=k}^{n} t_i \\ \text{if } x_k \le y < x_k + t_k \text{ for some } 1 \le k \le n \\ \text{or } x_k - \tau < y \le x_k \text{ for some } 1 \le k \le n-1. \end{cases}$$

In other words, in the absence of a collision, we have $F(x_1, x_2, \ldots, x_n, y) = x_n + t_n$. If a collision occurs during the execution of some job $w_k \in W$,

$$F(x_1, x_2, \ldots, x_n, y) = y + \tau + \sum_{i=k}^{n} t_i.$$

This means that the optimal guaranteeing strategy of the first player to schedule the execution of the basic job complex *W* is $x_1^0 = 0,\ x_i^0 = x_{i-1}^0 + t_{i-1}$, $i = \overline{2, n}$. In this case, all the basic jobs *W* will be completed at the time $x_n^0 + t_n$ (no collision) or $x_n^0 + t_n + \tau$ (collision occurrence). With any other strategy $x_1, x_2, \ldots, x_n$, in the worst case, the basic job complex *W* will be completed at the time $x_n + t_n + \tau >$ $x_n^0 + t_n + \tau$ since $x_n > x_n^0$. Thus, the strategy $(x_1^0, x_2^0, \ldots, x_n^0)$ is the optimal guaranteeing strategy under $m = 1$.

Consider an illustrative example.

Example 1. For $n = m = 1$,

$$F(x, y) = \begin{cases} x+t \\ \text{if } 0 \le y \le x-\tau, \\ \text{or if } x+t \le y \le T \\ \text{(no collision);} \\ \\ y+\tau+t \\ \text{if } x-\tau < y < x+t \\ \text{(collision occurrence).} \end{cases}$$

The graph of this payoff function is shown in Fig. 1.

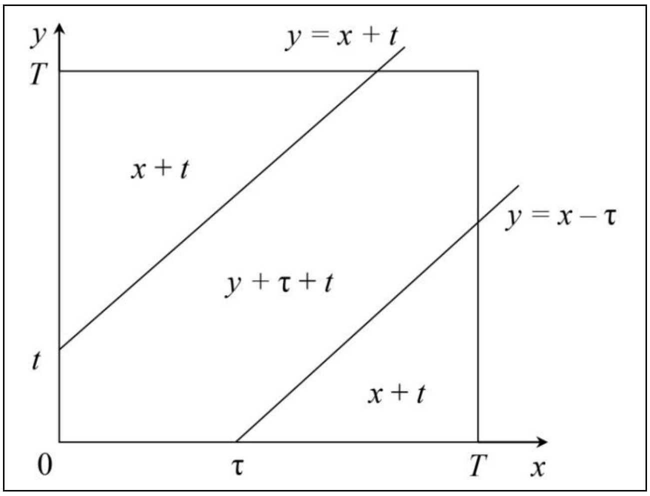

Fig. 1. The payoff function in Problem 1 with *n* = *m* = 1.

In this case, we have $F(0, y) < 2t + \tau$ for all $y \in [0, T]$ and $F(x, y) > 2t + \tau$ for $x \ne 0$ and any $y \in (x - \tau, x + t)$. Hence, $x^0 = 0$ is the optimal guaranteeing strategy of the first player. ♦

The case $n > 1, m > 1$ is considered by analogy. As a scheduling strategy we choose $\left(x_1^0, x_2^0, \ldots, x_n^0\right)$. In the absence of a collision on the interval $\left[x_1^0, x_1^0 + t_1\right]$, we fix x_1^0 and let $x_2^0 = x_1^0 + t_1$. If a collision occurs on the interval $\left[x_1^0, x_1^0 + t_1\right]$ at the time y_1, we let $x_1^0 = y_1 + \tau_1$. Similarly, in the first case, the interval $\left[x_2^0, x_2^0 + t_2\right]$ is examined for collisions; in the second case, the interval $\left[x_1^0, x_1^0 + t_1\right]$, and so on. Thus, the optimal job scheduling strategy W is constructed dynamically, depending on the arrival of supplementary job requests.

3. SOLUTION OF PROBLEM 2

Now we define the payoff function of the antagonistic game as follows:

$$F(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m) = \begin{cases} 1 \text{ if } x_i + t_i \le y_j \le x_{i+1} - \tau_j \\ \text{for all } i = \overline{1, n} \text{ and } j = \overline{1, m}, \\ \text{or } 0 \le y_j \le x_1 - \tau_j, \\ \text{or } x_n + t_n \le y_j \le T; \\ \\ 0 \text{ if } x_i \le y_j \le x_i + t_i \\ \text{for some } 1 \le i \le n,\ 1 \le j \le m, \\ \text{or } x_i - \tau_j \le y_j \le x_i. \end{cases}$$

Let $\bar{x} = (x_1, x_2, \ldots, x_n)$ and $\bar{y} = (y_1, y_2, \ldots, y_m)$. Then $F(\bar{x}, \bar{y}) = 1$ if there is no collision and $F(\bar{x}, \bar{y}) = 0$ otherwise. Let $X = \left\{\bar{x} : x_i + t_i \le x_{i+1},\ i = \overline{1, n-1},\ x_1 \ge 0,\ x_n \le T\right\}$, $Y = \left\{\bar{y} : y_j + \tau_j \le y_{j+1},\ j = \overline{1, m-1},\ y_1 \ge 0,\ y_m \le T\right\}$, and $f(\bar{x})$ be the mixed strategy of the first player, i.e., a probability measure on the set X. By the definition of the payoff function $F(\bar{x}, \bar{y})$,

$$E(f, \bar{y}) = \int_X F(\bar{x}, \bar{y})\, df(\bar{x})$$

is the probability of no collision under a fixed $\bar{y}$. The optimal mixed strategy of the first player, $f^0(\bar{x})$, maximizes the value of

$$\min_{\bar{y} \in Y} E(f, \bar{y}):$$

$$\max_f \min_{\bar{y} \in Y} E(f, \bar{y}) = \min_{\bar{y} \in Y} E\left(f^0, \bar{y}\right),$$

where the maximum is taken over all probability measures on the set X. In other words, $f^0(\bar{x})$ is the best guaranteeing strategy of the first player.

Example 2. Consider the case $n = m = 1$, $X = [0, 1]$, $Y = [0, 1]$, and $T = 1$. We will write $x, y, t,$ and τ instead $x_1, y_1, t_1,$ and τ_1, respectively. Let $t \le 0.25$ and $\tau \le 0.25$. Then

$$F(x, y) = \begin{cases} 1 \text{ if } y \ge x + t \\ \text{or } y \le x - \tau; \\ \\ 0 \text{ if } x - \tau < y < x + t \end{cases},$$

$$x \in [0, 1],\ y \in [0, 1].$$

The graph of the payoff function F is shown in Fig. 2.

Let $f^0(x)$ be a probability measure uniformly distributed on the interval [0, 1]. By the definition of the payoff function $F(x, y)$, we have

$$\int_0^1 F(x, y)\,df^0(x) \ge 1-\tau-t$$

for all $y \in [0, 1]$. If the probability measure $f(x)$ is not uniformly distributed on the interval [0, 1], then there exists a segment $[x_0, x_0+\tau+t] \subseteq [0, 1]$ such that

$$\int_{x_0}^{x_0+\tau+t} df(x) > \tau+t.$$

Hence, there exists a value $y \in [0, 1]$ for which

$$\int_0^1 F(x, y)\,df(x) < 1-\tau-t.$$

Therefore,

$$\int_0^1 F(x,y)\,df^0(x) = \max_f \min_{y\in[0,1]} \int_0^1 F(x,y)\,df(x),$$

where the maximum is taken over all probability measures on the interval [0, 1]. So, $f^0(x)$ is the optimal mixed strategy of the first player. ♦

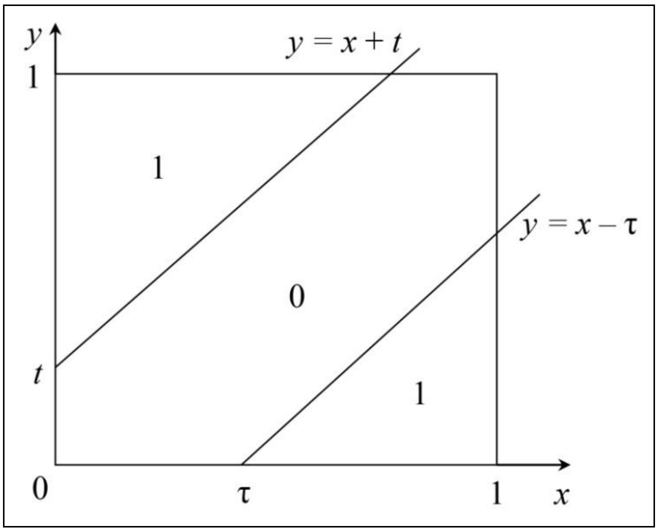

Fig. 2. The payoff function in Problem 2 with *n* = *m* = 1, *t* ≤ 0.25, τ ≤ 0.25, and *T* = 1.

Next, we can apply the discrete approximation method of an infinite game by a finite one [15]. For any $\varepsilon > 0$, this method yields the ε-optimal mixed strategy of the first player, $f_\varepsilon(\bar{x})$, concentrated in a finite number of points. Let $f_\varepsilon(\bar{x})$ be concentrated in points $v_1, v_2, \ldots, v_p$ with jumps

$$q_1,\ q_2, \ldots,\ q_p,\ q_j > 0,\ j = \overline{1,\ p},\ \sum_{j=1}^{p} q_j = 1.$$

Each point v_j, $j = \overline{1, p}$, is associated with some schedule for the job set W, which should be executed with probability q_j.

Example 3. Let $n = m = 1$, $X = [0, 1]$, $Y = [0, 1]$, $T = 1$, $t = 0.25$, and $\tau = 0.25$. The payoff function has the following form:

$$F(x,\ y) = \left\{ \begin{array}{c} 1 \text{ if } \ y \ge x + 0.25 \\ \text{or } y \le x - 0.25; \\ \\ 0 \text{ if } \ x - 0.25 < y < x + 0.25 \end{array} \right\},$$

$$x \in [0, 1],\ y \in [0, 1].$$

According to Example 2, $f^0(x)$ is the optimal mixed strategy of the first player and

$$\int_0^1 F(x,\ y)\,df^0(x) \ge 1 - 0.25 - 0.25 = 0.5$$

for all $y \in [0, 1]$. Let $\bar{f}(x)$ be a probability measure on the segment [0, 1] concentrated in two points, $x = 0$ and $x = 1$, with jumps of 0.5. Then

$$\int_0^1 F(x,\ y)\,d\bar{f}(x) \ge 0.5$$

for all $y \in [0, 1]$. Hence, like $f^0(x)$, $\bar{f}(x)$ is the optimal mixed strategy of the first player. Thus, the job w_1 should be started at time 0 or time 1 equiprobably (with probability 0.5).

The graph of the payoff function *F* is shown in Fig. 3.

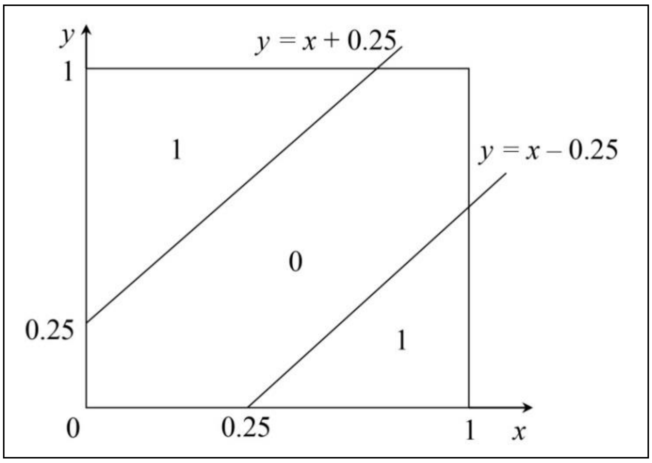

Fig. 3. The payoff function in Problem 2 with *n* = *m* = 1, *t* = 0.25, τ = 0.25, and *T* = 1.

4. THE PROBLEM WITH NON-FIXED JOB DURATIONS

In this section, we assume the availability of *L* types of additional non-renewable resources to execute the basic job complex *W*, in amounts $R_1, R_2, \ldots, R_L$, respectively. (Non-renewability means that the resources cannot be reused.) If a job $w_i \in W$ is allocated an amount r_{il} of the lth resource type, $l=\overline{1,\ L}$, its duration will be reduced to

$$t_i = t_i^0 - \sum_{l=1}^{L} a_{il} r_{il},\ i=\overline{1,\ n},$$

where a_{il} are given nonnegative numbers; t_i^0 denotes the duration of this job without additional resources allocated. By assumption, the following constraints hold:

$$0 \le r_{il} \le r_{il}^0,\ i=\overline{1,\ n},\ l=\overline{1,\ L}, \tag{1}$$

$$\sum_{i=1}^{n} r_{il} \le R_l,\ l=\overline{1,\ L}, \tag{2}$$

$$t_i^0 - \sum_{l=1}^{L} a_{il} r_{il}^0 > 0, \tag{3}$$

where $r_{il}^0,\ i=\overline{1,\ n},\ l=\overline{1,\ L},$ are given positive numbers (the maximum admissible amounts of resources that can be allocated to the job). Inequalities (1) limit the amounts of each resource type that can be allocated to each job. Next, inequalities (2) limit the total amount of each resource type allocated to all jobs together. Finally, inequalities (3) limit the durations of the jobs. A resource allocation $r_{il},\ i=\overline{1,\ n},\ l=\overline{1,\ L}$, is called feasible if conditions (1)–(3) are valid.

The objective is to determine a feasible resource allocation facilitating the solution of Problems 1 and 2. According to Sections 2 and 3, such a resource allocation minimizes the total duration of the job complex *W*. Thus, we arrive at the following linear programming problem:

$$\min_{r_{il},\, i=\overline{1,n},\, l=\overline{1,L}} \sum_{i=1}^{n} \left(t_i^0 - \sum_{l=1}^{L} a_{il} r_{il} \right)$$

subject to the constraints (1) and (2). The solution of this problem will give the optimal feasible resource allocation.

CONCLUSIONS

In this paper, we have studied the scheduling problem for a complex of basic jobs under the condition that at some uncertain times, execution requests for supplementary higher-priority jobs are received. The execution sequence of the basic jobs is fixed. If a supplementary job request arrives during the execution of a basic job, then the latter is terminated and must be restarted at some time upon the complete service of the former. All jobs (basic and supplementary) are executed without interruption. Two problem formulations have been considered. In the first, the performance criterion is the completion time of the basic job complex, and the problem is to minimize this time. In the second formulation, the probability of a collision is minimized, as a situation where a supplementary job request arrives during the execution of a basic job. The problems have been solved via their reduction to infinite zero-sum two-player (antagonistic) games and the discrete approximation of the latter by finite games. The scheduling method has been illustrated on model examples. Also, the problem formulation with non-fixed durations of the basic jobs, linearly dependent on the amount of additional resources allocated, has been investigated. In this case, a feasible schedule is found by solving a linear programming problem.

The results of this paper can be used to plan computations during the testing and operation of complex technical objects (such as airplanes and nuclear reactors). In the planned mode, computations are performed using application modules, and an abnormal situation may occur at uncertain times (e.g., the values of some parameters may go beyond an admissible range). In this case, scheduled computations are interrupted and supplementary higher-priority jobs are executed.

Scheduling problems under uncertainty were studied in [8–10] under the assumption of renewable resources and the non-fixed values of some parameters (such as job durations or the amounts of available resources). The parameters were defined through either their admissible ranges or probabilistic characteristics. The solution algorithms were based on the branch-and-bound method. In contrast to the cited works, this paper has addressed a scheduling problem with uncertain request arrivals. Also, the case of additional non-renewable resources has been investigated. Problems with a heterogeneous set of resources were considered in [13, 14] in the deterministic setup.

In the future, we intend to analyze a more general problem formulation with several computing devices for basic and supplementary jobs.

REFERENCES

1. Kononov, D.A., Security Research of Control Systems Based on the Analysis of Their System Parameters, *Trudy 28-oi Mezhdunarodnoi konferentsii "Problemy upravleniya bezopasnost'yu slozhnykh sistem"* (Proceedings of the 28th International Conference on Problems of Complex Systems Security Control), Moscow, 2020, pp. 102–108. (In Russian.)
2. Kononov, D. and Furugyan, M., Control of a Complex of Works in Multiprocessor Real-Time ACS, *Proceedings of the 1st International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA2019)*, Lipetsk, 2019. URL: https://ieeexplore.ieee.org/document/8947570.
3. Tanaev, V.S., Gordon, V.S., and Shafranskii, Ya.M., *Teoriya raspisanii. Odnostadiinye sistemy* (Scheduling Theory: Single-Stage Systems), Moscow: Nauka, 1984. (In Russian.)

4. Brucker, P., *Scheduling Algorithms*, Heidelberg: Springer, 2007.
5. Lazarev, A.A., *Teoriya raspisanii. Metody i algoritmy* (Scheduling Theory. Methods and Algorithms), Moscow: Trapeznikov Institute of Control Sciences RAS, 2019. (In Russian.)
6. Glonina, A.B. and Balashov, V.V., On the Correctness of Real-Time Modular Computer Systems Modeling with Stopwatch Automata Networks, *Modeling and Analysis of Information Systems*, 2018, vol. 25, no. 2, pp. 174–192. https://doi.org/10.18255/1818-1015-2018-2-174-192 (In Russian.)
7. Glonina, A.B., Tool System for Testing Real-Time Constraints for Modular Computational System Configurations, *Moscow Univ. Comput. Math. Cybern.*, 2020, vol. 44, no. 3, pp. 120–132. https://doi.org/10.3103/S0278641920030036
8. Koshelev, P.S. and Mishchenko, A.V. Optimizing Management of Jobs in a Logistic Project under Conditions of Uncertainty, *Journal of Computer and Systems Sciences International*, 2018, vol. 60, no. 4, pp. 595–609.
9. Gorskii, M.A., Mishchenko, A.V., Nesterovich, L.G., and Khalikov, M.A., Some Modifications of Integer Optimization Problems with Uncertainty and Risk, *Journal of Computer and Systems Sciences International*, 2022, vol. 61, no. 5, pp. 813–823.
10. Kosorukov, O.A., Lemtyuzhnikova, D.V., and Mishchenko, A.V., Methods and Models of Project Resource Management under Uncertainty, *Journal of Computer and Systems Sciences International*, 2023, vol. 62, no. 2, pp. 305–323.
11. Yao, X., Almatooq, N., Askin, R.G., and Gruber, G., Capacity Planning and Production Scheduling Integration: Improving Operational Efficiency via Detailed Modelling, *Intern. J. Production Research*, 2022, vol. 60, no. 1, pp. 7239–7261.
12. Graves, S.C., How to Think About Planned Lead Times, *Intern. J. Production Research*, 2022, vol. 60, no. 1. DOI: http://dx.doi.org/10.2139/ssrn.3485059
13. Kononov, D.A., Furugyan, M.G. Planning a Complex of Works with Heterogeneous Resources under Uncertainty, *Proceedings of 2022 15th International Conference on Management of Large-Scale System Development (MLSD)*, Moscow, 2022. DOI: 10.1109/MLSD55143.2022.9934381
14. Kononov, D.A. and Furugyan, M.G., Effective Means of Regional Management: Optimal Use of Heterogeneous Resources, *Proceedings of 2021 14th International Conference on Management of Large-Scale System Development (MLSD'2021)*, Moscow, 2021. DOI: 10.1109/MLSD52249.2021.9600251
15. Furugyan, M.G., On Approximating the Solution of a Certain Class of Infinite Antagonistic Games, *Vestn. Mosk. Gos. Univ. Ser. 15. Vych. Mat. Kibern.*, 1978, no. 2, pp. 81–85. (In Russian.)

This paper was recommended for publication by A. A. Lazarev, a member of the Editorial Board.

Received April 4, 2025,
and revised June 30, 2025.
Accepted July 16, 2025

Author information

Kononov, Dmitry Alekseevich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ dmitrykon52@gmail.com
ORCID iD: https://orcid.org/0000-0002-6059-5590

Furugyan, Meran Gabibullaevich. Cand. Sci. (Phys.–Math.), Federal Research Center "Computer Science and Control," Russian Academy of Sciences, Moscow, Russia
✉ rtsccas@yandex.ru
ORCID iD: https://orcid.org/0000-0003-0373-9233

Cite this paper

Kononov, D.A. and Furugyan, M.G., Managing a Complex of Jobs with Uncertain Execution Request Arrivals. *Control Sciences* **4,** 25–31 (2025).

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com