OPTIMIZING TASK ASSIGNMENT AMONG UNMANNED VEHICLES IN TERMS OF ENERGY CONSUMPTION

K. V. Semenkov*, E. A. Abdulova**, E. F. Jharko***, and V. G. Promyslov****

*–****Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

*✉ semenkov@ipu.ru, **✉ consoft@ipu.ru, ***✉ zharko@ipu.ru, ****✉ vp@ipu.ru

Abstract. This paper considers the topical issue of ensuring the availability of unmanned vehicles (agents) in a dynamic technical system (DTS) within an intelligent transport environment. The problem of uneven load distribution among agents, causing inefficient energy consumption and reducing the total operating time of the system, is studied. This problem is solved by proposing an optimization model that includes an objective function (maximizing the total operating time of the entire DTS) and a set of constraints describing the available energy of each agent. The key aspect of the model is to ensure the uniform distribution of the energy load among all agents. The optimization problem is solved using the CP-SAT Boolean Satisfiability algorithm with integer constraints. According to the experimental results with the CP-SAT algorithm, there is an interesting phenomenon, i.e., a correlation between the sampling step (the time interval during which the algorithm searches for an acceptable solution) and the execution time of the optimization program. Based on this correlation, a heuristic method for changing the sampling step is proposed. The study is primarily focused on the performance of the model and optimization algorithm in real conditions of robotic transport systems with exogenous disturbances. According to the testing results, the model demonstrates good performance on virtual agents with completely known system parameters and on a group of real agents (wheeled robots), where the system parameters are subject to disturbances.

Keywords: dynamic technical system, availability, optimization, distribution of energy resources, unmanned vehicle, safety.

INTRODUCTION

The development of models and methods for improving the efficiency of a group of autonomous devices (agents) in a dynamic technical system (DTS) within an intelligent transport environment is an important field in robotics. This includes the current tasks of controlling the energy resources of individual agents and the entire group: the efficiency of task fulfillment, functionality, and autonomous operation time of the system directly depend on its energy and the scenarios for using this energy. The technical feasibility of reassigning tasks or exchanging resources between agents in a group motivates researchers to develop approaches where the energy reserve and tasks fulfilled by an individual agent within a DTS are treated as a single distributed asset.

One area of application for such approaches could be intelligent DTSs involved in the industrial process of an enterprise. Their important characteristics are functional safety and availability. According to GOST R IEC 61508-4-2012 [1], functional safety is “part of the overall safety relating to the EUC [*equipment under control*] and the EUC control system which depends on the correct functioning of the E/E/PE [*electrical/electronic/programmable electronic*] safety-related systems, other technology safety-related systems and external risk reduction facilities.” Availability is “the ability of an object to perform the required functions under specified conditions, at a specified time or during a specified period, given that all necessary external resources are provided”; see GOST R 27.102-2021 [2]. In this paper, an inoperable state of an unmanned vehicle (UV) is understood as a state in which it cannot fulfill tasks due to energy depletion.

Availability directly affects the completeness of DTS safety [3]. Let us explain this statement with an example. Suppose that the energy of a UV fulfilling a task in a robotic transport system of an enterprise is close to zero (e.g., the batteries are almost discharged). In this case, the UV becomes inoperable (unavailable), and one should urgently remove all tasks from it and transfer it to a safe state, since an abnormal termination of the task may have dangerous consequences.

On the other hand, for economic reasons, it is desirable that the downtime of a UV in the system be minimal, i.e., the DTS operation scenarios should be aimed at increasing the utilization factor [2] of vehicles. In this paper, the downtime of a UV is understood as a state in which the vehicle cannot fulfill tasks due to energy depletion or because it has not been assigned a suitable task in terms of resource utilization. Thus, the energy-efficient distribution of tasks among mobile agents in the DTS of an enterprise is important, both for ensuring the safety of the DTS and for increasing the economic efficiency of the entire enterprise.

Depending on the problem specifics, it can be formulated as a mathematical assignment problem, a knapsack problem, a bin packing problem, etc. The problem belongs to the class of NP-complete ones. Algorithms for solving such problems are known (e.g., dynamic programming algorithms); given a definite set of constraints and an objective function, the algorithms yield a solution in a reasonable time (if the solution exists), considering the agent's resources and constraints on the problem parameters.

Current research is mainly focused on specifying the type of objective functions and constraints, as well as on selecting effective optimization algorithms. For example, optimal task assignment among mobile robots in order to optimize the charge-discharge cycle of individual agents was investigated in [4], taking into account weather conditions and their routes. The model developed therein includes such parameters as velocity, acceleration, and payload. Gradient methods for global optimization [5] and a heuristic "auction procedure" for local optimization were used as optimization algorithms. As concluded by the authors, the local approach based on predicting operating time and energy reserve has higher flexibility and is less computationally intensive. Hence, this approach is preferable for complex scenarios. The energy consumption of a robotic cell system in an enterprise was considered in [6]. The model covers the velocity of the robots, their placement within the cell, the energy-saving modes supported, and possible operation chains. The optimization problem was solved using mixed-integer linear programming (MILP) and hybrid heuristics for large systems based on the Gurobi library [7]. According to the experiments conducted on a real system, energy consumption can be reduced by 20% by optimizing the sequence of operations and applying energy-saving modes. A simulation model of a group of robots fulfilling tasks with the redistribution of energy resources between them was studied in [8]. The model was transformed into an assignment problem with an objective function that minimizes the total task fulfillment time in the form of a linear norm; the Hungarian algorithm [9] was applied for optimization.

In addition, there are sufficiently many surveys of the subject, which emphasize the mass scale and topicality of energy optimization. Methods for improving energy efficiency in robotic and mechatronic systems were systematically reviewed in [10]. The main attention was paid to reducing the dimension of optimization problems by using approximate models. As noted, in most real systems, it is impossible to obtain an exact solution of the optimization problem.

The above review of methods for ensuring the energy efficiency of robotic systems was complemented in [11], albeit with the focus on objective functions and the expected gain of optimization rather than on the mathematical aspects of solving the corresponding optimization problem. The authors described in detail methods for optimizing robotic systems and temporal and sequential planning approaches to minimize energy consumption. Particular attention was paid to issues of controlling the robot's hardware and software environment and the need for their joint analysis to improve energy efficiency.

All publications emphasize the importance of choosing an adequate model for applying mixed-type algorithms with heuristics (which reflect the structure of the object or objective function) in order to reduce the search space when solving the optimization problem.

This paper considers optimization issues for energy consumption in a DTS, where UVs fulfill a certain set of tasks. Tasks are assigned to DTS agents by a control center, with which UVs exchange information. UVs operate independently of each other. Each task requires certain UV resources (e.g., load capacity for transporting cargo or tank volume for transporting liquids). A resource characterizes the task but not the UV fulfilling the latter, and the UV resources involved in the task are independent. The system is open in terms of the number of agents, tasks, and resources, i.e., the number of UVs and tasks in the system and the UV resources available to fulfill a task at the next discrete time instant may change. (An example of a change in a UV resource is the installation of a tank of a different volume.)

We proceed to another important assumption. Let the DTS under consideration be compact, i.e., the characteristic scale of the system is much smaller than the path traveled by the UVs during operation. Then it is possible to neglect the issues related to the spatial location of agents (e.g., the implementation of task transfer between UVs) and focus on various static constraints on task fulfillment by an agent.

The objective function in an optimization problem depends on the system usage scenario, and this paper considers a scenario aimed at maximizing the utilization factor of the DTS.

Theoretical approaches to solving such problems are well-known, but they are often tested only on computer models; at the same time, the behavior of a real system can significantly differ from an idealized computer model. The goal of this work is to test known optimization algorithms on a real DTS model, i.e., to develop a system energy consumption model that is applicable to a wide range of DTS agent parameters and requires no special tuning for each agent.

As underlined in the literature, the MILP algorithm [12] is an effective optimization algorithm for such problems. In this paper, we use the MILP implementation in the OR-TOOLS library [13]. The MILP algorithm is supplemented with a heuristic block that estimates the sampling step based on indirect characteristics; according to the authors' observations, this approach reduces the amount of computations and interference in the system operation. The constraints are the minimum and maximum values of resources consumed when fulfilling tasks on DTS agents, the "affinity" of tasks with each other, and restrictions on the launch of tasks on certain agents. The objective function is a concave (quadratic) function with a restriction that the variables are integers [14].

In addition to the main problem (optimizing energy consumption), the partial problem solved is to distribute tasks among DTS agents during the initial task assignment. In this case, to accelerate computations, some constraints may be discarded, or optimization may be skipped; as a result, the problem can be solved using linear programming methods.

This paper considers a mathematical model for controlling a group of DTS agents (ground robots) with task reassignment among them. The control model ensures the system's availability for a long time by optimizing the energy consumption of individual UVs and maximizing the availability factor. The model takes into account constraints on task fulfillment, both in terms of fundamental feasibility and in terms of the necessary resources available to UVs.

A distinctive feature of this work is that the mathematical model is tested not only on virtual agents but also on an experimental mock-up, i.e., a group of wheeled robots. This approach allows assessing the applicability of the model for practical needs and identifying the characteristics of real DTSs that should be included in the model.

1. PROBLEM STATEMENT AND SYSTEM MODEL

When building a DTS model, various requirements may be imposed on it, depending on both the object and the experience of the model developer. But the main requirement should be the model's capability to reflect the intra-system interaction of those agents that influence the energy characteristics of the system [15]. The model presented below follows this principle.

1.1. Basic Provisions

Let a DTS Ξ consist of a finite set of agents $S=\{s_i\}$, $i=1,\ldots,B$, which fulfill a certain set of tasks $\Phi=\{\phi_l\}$, $l=1,\ldots,N$, at each time instant. To fulfill these tasks, the agents have certain divisible resources of different types at their disposal: $R=\{r_{ij}\}$, $r_{ij}\geq 0$, $i=1,\ldots,B$, $j=1,\ldots,M$, where i and j denote the agent's number and the resource type, respectively. In the general case, for each set, several static constraints can be defined on the admissible range D for the system.

A definite amount of resources of different types is needed to fulfill each task, which is described by a mapping

$$Q(\phi):\Phi\rightarrow R. \tag{1}$$

The tasks are reassigned in the system using some impact (control) $U=\{u_i\}$, $i=1,\ldots,N$, so for clarity, we will equip the mapping of the tasks set Φ into the agents set S with the subscript u:

$$L_u(\phi):\Phi\rightarrow S. \tag{2}$$

The mappings Q and L_u may include definite static constraints on resource use and task assignment.

At any time instant, the state of this system is given by the tuple

$$\Xi=\langle S, R, Q, L_u\rangle. \tag{3}$$

We will consider the memoryless system, i.e., a system where the mapping $L_u(\phi)$ is determined only by the current state of the system, regardless of the history of transitions to this state.

Given an initial state Ξ_0 of the system, optimal control consists in selecting such an impact U that will optimize an objective function of certain phase coordinates (parameters) of the system. As the only parameter of the system, we will take the vector $E=\{e_i\}$, $i=1,\ldots,N$, which represents the energy of this system. The energy of a UV in a physically realizable system is a smooth or piecewise smooth function; therefore, it is possible to pass from continuous to discrete time, compute the energy on a certain grid, and then approximate the result.

Let the system states at time instants $k+1$ and k be related by

$$\Xi_{k+1}=F_D\left(\Xi_k, U_{k+1}\right), \quad (4)$$

where the subscript D means that the constraints are taken into account.

We define a mapping

$$g(\Xi)\to E, \quad (5)$$

which allows computing the change in the system energy under a known task assignment among agents.

By assumption, given the state of the system Ξ at time instant k, its state at time instant $k+1$ is described by the difference equation

$$E_{k+1}=E_k+\alpha g\left(\Xi_k\right),\ 0\le\alpha\le 1. \quad (6)$$

We introduce a control efficiency function of the form

$$f(E_k,\ U_{k+1}). \quad (7)$$

At each time instant, an optimal decision is made to transition the system to the next state, and the overall control optimality criterion is calculated as the sum of the ones at each time instant. Then the optimization is additive, and the objective function takes the form

$$Z=\sum_{i=1}^{k}\min_D\left(f\left(E_i, U_{i+1}\right)\right). \quad (8)$$

According to the operating scenario (see the Introduction), the utilization factor of UVs should be high enough; therefore, control should distribute the load evenly among all UVs in the system. The load is calculated as the ratio of the agent's energy to the energy consumed for all tasks assigned to the agent. In this case, it is logical to take the minimum energy variance across the agents in the system as an estimate of the control efficiency f. This is defined by a quadratic form of the objective function, ECE^{T}, with a symmetric matrix $C\succeq 0$ of dimensions $n\times n$. For a sufficiently small α, the objective function has a unique fixed point to which the solution will converge [16], and the convergence rate is linear.

Owing to the assumption of independent tasks and resource types, optimization can be carried out separately for each resource type.

In most real cases, this optimization problem cannot be solved analytically [17], and there are also known difficulties in assessing the stability and convergence of the algorithm due to the presence of disturbances. Nevertheless, as practice shows, numerical optimization methods quickly yield a very small value of the error function [18, *p. 153*].

There exist established algorithms for the numerical solution of such problems; some of them were overviewed in [19, 20]. To solve the optimization problem of the above class, in this paper, we choose the well-proven MILP method [13] with heuristics considering the features of the controlled object and the objective function.

1.2. Model Description

Using the formalism introduced by equations (1)–(8), we consider the system Ξ of UVs capable of fulfilling certain tasks. The UVs in the model are not supposed to be homogeneous: each vehicle has individual technical characteristics. Additional constraints may be imposed on resources and UVs intended for fulfilling certain tasks, i.e., not all UVs are equally suitable for fulfilling specific tasks. In real UVs, constraints can be both quantitative (e.g., each UV has an individual maximum load capacity) and qualitative, related, e.g., to the absence of a particular sensor or equipment on the UV. In the method developed, constraints can be considered by specifying an admissible range D for resources and tasks (see Table 1). By assumption, each task is characterized by an individual set of resources consumed.

At the initial time instant, each UV (agent) has a certain initial energy reserve (e.g., battery charge); during the DTS operation, this reserve does not increase (i.e., batteries are not recharged or replaced). It is necessary to assign tasks among UVs by maximizing the time during which all UVs will have a positive energy reserve.

Suppose that there are B UVs and N tasks. Unless otherwise specified, the subscripts b and i will denote the UV number and the task number, respectively.

We introduce the following notation:

- $E_b(t)$ is the UV's energy at time instant t. Since no energy comes to the system, $E_b(t)$ is a non-increasing function.
- $h(i, b, t) = \begin{cases} 1 \text{ if task } i \text{ is fulfilled} \\ \quad \text{by UV} b \text{ at time instant} t \\ 0 \text{ otherwise.} \end{cases}$
- p_i is the specific power (per unit of resource) consumed when fulfilling task i. For simplicity, assume that p_i is independent of the resource value.
- q_{ib} is the current consumption of a given type of resource on the UV when fulfilling the task i .

Let us divide the operating time of the system into intervals $\tau_k = t_{k+1} - t_k$, $k = 1, \ldots, K$, so that during each interval, the energy consumed to fulfill task i is described by the simple linear relation [21]

$$\Delta e_{ibk} = -p_i q_{ib}(t_{k+1} - t_k).$$

During the time interval τ, the change in the energy of an individual UV can be written as

$$E_b(t_{k+1}) - E_b(t_k) = \sum_{i=1}^{N} \Delta e_{ibk} h(i, b, t_k),$$
$$k = 1, \ldots, K,$$

and the change in the energy of the entire DTS during the time interval τ takes the form

$$E(t_{k+1}) - E(t_k) = \sum_{b=1}^{B} \sum_{i=1}^{N} \Delta e_{ibk} h(i, b, t_k),$$
$$k = 1, \ldots, K.$$

To define the mappings (1) and (2), we introduce the following additional designations:

- r_{ib} is the minimum resource value required to fulfill the task (e.g., the minimum load that the UV must carry or the background load of computing units).
- R_{ib} is the maximum resource value when fulfilling the task.
- C_b is the UV resource specifications (load capacity, onboard processor frequency, etc.).
- $\kappa_{ib} = 0.8C_b$ is the setting for a high load level.
- $\lambda_{ib} = 0.1\, C_b$ is the setting for a low load level.
- $A(q_{ib}) = H(\lambda_{ib} - q_{ib}) + H(q_{ib} - \kappa_{ib})$ is the setting function, where $H(x)$ indicates the Heaviside function.
- S_i is the coefficient reflecting the priority of the task.

Table 1 presents the constraints D on the distribution of resources and tasks among agents.

As mentioned above, the objective function is convex, but for a real system it has a more complex form and includes a set of Boolean functions depending on the system parameters (Table 2).

Table 1

The list of constraints implemented in the model

No.	Constraint	Comment
1	$\sum_{i=1}^{N} \Delta e_{ibk} h(i, b, t_k) \le E_b(t_k)$	The energy consumed by all tasks during time τ does not exceed the agent's available energy $E_b(t)$.
2	$\sum_{i=1}^{N} r_{ib} \le C_b$	The UV's characteristics allow the task to be fulfilled. The minimum resources required by all tasks assigned to the agent do not exceed the latter's available resources. Without this constraint, the system may have "overloading" during operation.
3	The constraint on the number of agents involved in task reassignment at each time instant, expressed as a percentage of the total number of agents in the DTS	–
4	The constraint on the type of tasks that a given agent can fulfill	The list of agents that can or cannot fulfill the specified task
5	The constraints on the joint fulfillment of tasks	For each task, a list of other tasks can be specified with which it must/must not be jointly fulfilled.

Table 2

The list of partial objective functions of the model

Partial objective function	Comment	No.
$\sum_{i=1}^{N} h(i, b, t)\left(\frac{R_{ib}}{C_b} + S_i\right) \to \min$	For all agents *b*, minimize the risk that if all tasks require the maximum possible resource value, the resource will not be enough for UVs considering the priority of the task	(9)
$D\left(E_b\left(t_j, \tau\right)\right) \to \min$	Minimize the energy consumption variance	(10)
$\sum_{i=1}^{N} h(i, b, t)\frac{r_{ib}}{C_b} \to \min$	Strive to assign tasks with greater resource consumption to agents with a higher reserve of that type of resource	(11)
$\mathrm{A}\left(q_{ib}\right) \to \min$	Avoid overloading an agent above level 1 and keep it loaded below level 2	(12)
$\sum_{i=1}^{N}\sum_{b=1}^{B}\left(1 - h(i, b, t)\right) \to \min$	Avoid keeping completely unused agents.	(13)
$\sum_{b=0}^{B} R_b^j \to \min$	Avoid reassigning tasks that consume significant resources, as such a reassignment can be difficult in practice. The constraint implicitly considers the fact that it is generally impossible to instantly remove a task from one agent and assign it to another. The duration and "cost" of task reassignment may depend on the type of task, the method of task transfer, etc.	(14)

The general objective function (8) for the DTS model has the form

$$Z = \left(\sum_{i=0}^{N} v_i \min\nolimits_{\arg} F_i\right), \tag{15}$$

where F_i are the partial optimization problems from Table 2; v_i is the weight for the partial optimization problem under all the constraints from Table 1; finally, $\arg$ means the parameters of optimization.

1.3. An Energy Consumption Optimization Procedure for the DTS

Finding an exact solution of the optimization problem (15) is not mandatory; we use an approximate solution obtained in a definite time instead. The algorithm for solving this problem is iterative since the system Ξ being optimized is dynamic in its nature. However, by assumption, changes in the system—e.g., the emergence of new tasks or the termination of those fulfilled previously—are discrete. The sampling step is selected depending on the rate of change of the parameters.

The partial objective functions and constraints specified in Tables 1 and 2 are considered or not, depending on the use scenario of the system being optimized.

In general terms, the solution can be described by the following sequence of actions.

Step 1. Initialization and initial assignment of tasks to DTS agents. The initial task assignment algorithm may neglect some constraints and objective functions to accelerate the start of the system. With "fast" initialization, all constraints from Table 1, except for the third, and the objective function (9) are taken into account. In this form, the optimization problem is a special case of the knapsack problem, which is often solved using integer linear programming methods. Some solution methods for such problems were presented in [12, 22, 23].

If we neglect some partial objective functions and, accordingly, avoid full optimization, limiting ourselves to a suitable solution that satisfies the constraints from Table 1 and the objective functions (12) and (13), the assignment problem will reduce to a system of linear equations.

Step 2. The main step of the algorithm. It consists in optimizing task assignment among agents under all constraints (Table 1) and the objective functions (9)–(14) (Table 2) of the system. The objective function (10) is quadratic and has the greatest weight in the general objective function, i.e., the total value represents a nonnegative definite quadratic form. For practical reasons (e.g., due to the presence of disturbances in the system), it is unnecessary to obtain an exact solution of the optimization problem; therefore, we use the *Boolean Satisfiability Problem* (SAT) algorithm [24]. It yields an acceptable solution for objective functions with constraints.

Step 3. Checking the algorithm's termination criterion (the arrival of new tasks to be assigned among agents and changes in the number of agents in the system). If new tasks appear or the number of agents changes, then the algorithm returns to Step 1; otherwise (the composition of agents or tasks remains the same), it returns to Step 2.

The termination criterion depends on the objective function specified. For example, if all agents must be preserved in the system, the termination criterion will be the depletion of energy for at least one agent; if the goal is to maximize the utilization factor of UVs, the termination criterion will be the lack of sufficient energy to fulfill all tasks at the next time instant.

At all steps of the general algorithm, the CP-SAT library [25] is used to solve the Boolean Satisfiability Problem. To simplify the optimization problem at Step 1 of the algorithm and meet the requirements for using the library [25], the quadratic function is linearized by introducing auxiliary variables and constraints. A similar approach was described in [26, 27].

2. MODEL'S PERFORMANCE ASSESSMENT

Assessing the performance of the residual energy control model for the DTS represents a challenge, as the performance criteria for real DTSs are largely subjective [17]. To evaluate the quality of this model, test scenarios were implemented both for virtual agents (mainly to assess control performance in known "ideal" conditions) and for an experimental mock-up, i.e., a group of real agents (wheeled robots) with disturbances and parametric uncertainties.

The initial assessment was carried out for groups of virtual agents in two trivial scenarios to easily design optimal control in terms of a given objective function and verify the correctness of the model.

Example 2.1. The DTS consists of homogeneous agents, all tasks are identical, and the number of tasks is a multiple of the number of agents. Obviously, the optimal solution for ensuring even system load with the maximum system lifetime is the uniform task assignment among agents.

Example 2.2. The DTS consists of agents, one of which has significantly higher energy resources than the others. All other agents are homogeneous, all tasks are identical, and the number of tasks is greater than the number of agents. The optimal solution for ensuring even system load with the maximum system lifetime is to load the agent with the highest energy reserve as much as possible and assign the remaining tasks evenly among the agents until equalizing the residual energy of the agents. When the system reaches this state, the problem will be reduced to that of Example 2.1. ♦

For all trivial cases, the algorithm proposed and the program developed demonstrated good performance. The solution was found in a reasonable time, determined by the characteristic time constant of the system, during which the system (presumably) remained stationary.

After the model's tests on trivial examples, complex tasks were considered. The parameters of the corresponding examples and their discussion are given below. Both the virtual and physical agents were taken in the DTS.

In all examples, the termination criterion was the system's incapability to fulfill the required tasks due to a lack of energy.

2.1. Algorithm's Performance Assessment Using Virtual Agents

Example 2.3. This example is a complication of Example 2.1. There are five agents (DTS1–DTS5) and 15 tasks (VM0–VM14). In this system, one of the agents (DTS5) has a significantly higher energy reserve compared to the others (by approximately an order of magnitude higher), while the energy reserves of the latter vary, with a spread not exceeding 400% in absolute value. The energy and resource demands for fulfilling the tasks are also different: tasks VM5 and VM7 consume approximately 20 times more energy than the others. In general, according to an expert estimate, the solution for ensuring uniform system load and maximizing its lifetime is to load agents with high energy reserves as much as possible; agents with low energy reserves should have a small load.

Figures 1 and 2 show task assignment among agents and the ratio of agents' resources to the tasks assigned to them as a result of control using the objective function (15). Here, as well as in Figs. 3 and 4, the free energy reserve of an agent is indicated by "Free resource." Obviously, the consumption of the first and second types of resources has an uneven ratio for different tasks, complicating optimization.

According to Figs. 1 and 2, in general, most of the energy-intensive tasks are assigned to the agent with the highest energy reserve. The constraint is the availability of the resources necessary to fulfill the task (constraints 2–5 in Table 1). For example, agent DTS5, which has a significant energy reserve, is assigned task VM7 (see Fig. 1), which requires the largest energy consumption and satisfies the resource constraints.

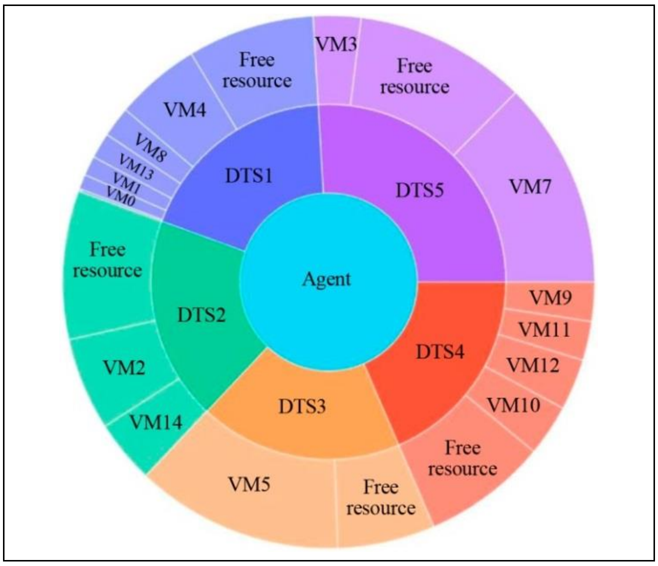

Fig. 1. Task assignment among agents for the first type of resource. All resource values are normalized by the resource maximum.

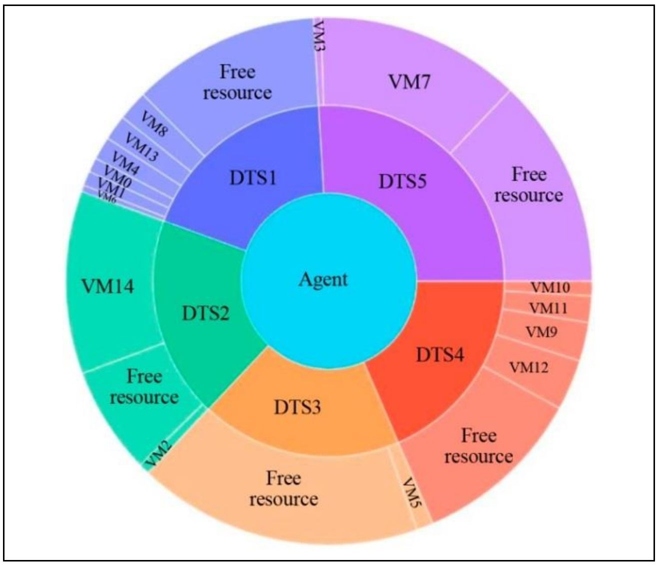

Fig. 2. Task assignment among agents for the second type of resource. All resource values are normalized by the resource maximum.

Next, Figs. 3 and 4 show the ratio of agents' energy to that required for fulfilling the tasks at the start of the system operation (a full energy charge) and at the final stage (energy depletion).

Finally, the energy dynamics of the system agents are presented in Fig. 5.

Obviously, owing to the control of task assignment, the agents' energies are gradually equalized, except for DTS1 (see label 1 in Fig. 5). The control effect for "equalization" is evident for agent DTS5, which initially has the highest energy reserve. This control strategy is optimal in terms of ensuring the maximum utilization of agents throughout the entire lifetime of the system.

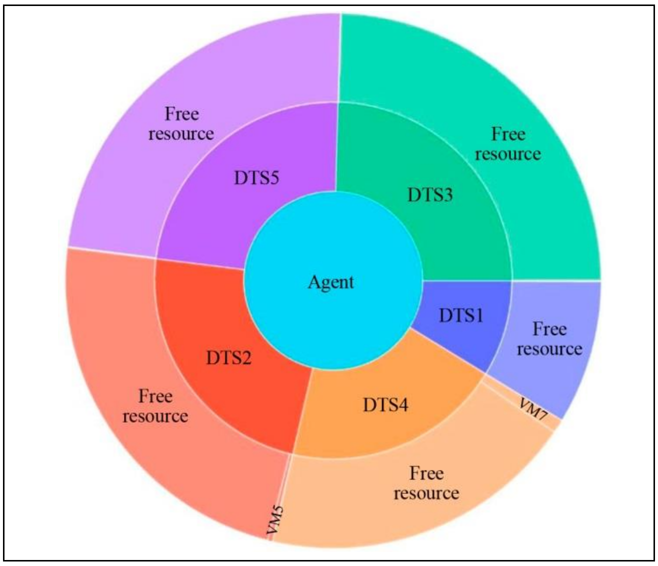

Fig. 3. Task assignment among agents at the start of the system operation. All resource values are normalized by the resource maximum.

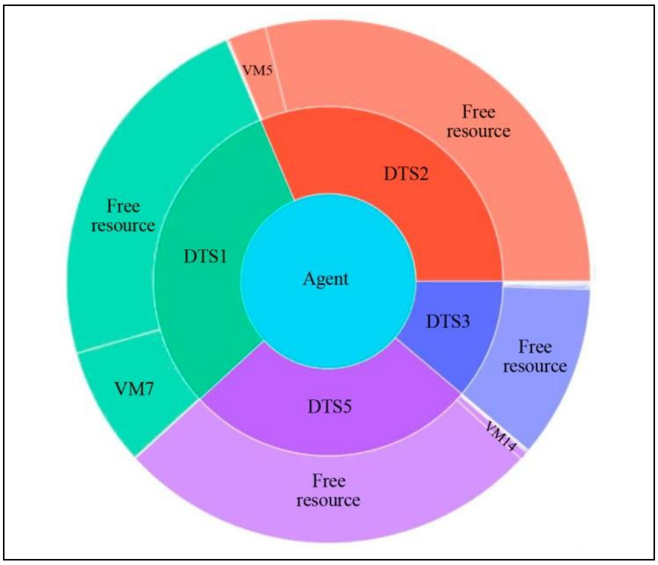

Fig. 4. Task assignment among agents at the final stage of the system operation (energy depletion). All resource values are normalized by the resource maximum.

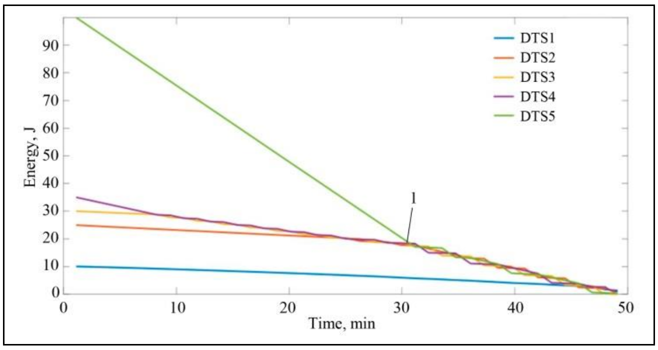

Fig. 5. The energy dynamics of system agents during operation. The horizontal axis corresponds to the time of day whereas the vertical axis to the UV energy (in J). The sampling step is 60 s.

2.2. Algorithm's Performance Assessment on Experimental Mock-Ups

The behavior and performance of the algorithm were also analyzed on an experimental mock-up of robotic systems. As a rule, control ideas and algorithms are mainly tested using virtual agents. There are many models that have been validated only on computer simulators [21, 28, 29]. Obviously, virtual agents do not provide a complete picture of all physical processes in a system, but due to the high cost and complexity, only a small part of the previous R&D work could be tested on experimental mock-ups [30]. The current stage in the development of robotic systems is characterized by cost reduction, mass supply, and high availability of components for creating intelligent robotic DTSs. Therefore, a significant amount of tests can now be carried out in laboratory conditions using comparatively cheap experimental mock-ups, as demonstrated in this work.

The block diagram of a real agent (wheeled robot) is provided in Fig. 6, and its components are specified in Table 3. A general view of the robot can be found in Fig. 7. The experimental mock-up consists of five robots (Fig. 8). The system's size was determined not by the computational limitations of the model or the laboratory's conditions but by the ultimate goal (an expert estimate of control efficiency): for a human, it is difficult to test a larger number of UVs. The robots were designed on the ESP32 platform and controlled by a central computer (including data processing from all robots and task assignment among them).

Table 3

The list of robot's components in the experimental mock-up of the DTS

Designation in Fig. Fig.	Purpose	Model
M_1 and M_2	Actuating electric motors	TT motor 3–6 V
U_1	Ultrasonic sensor	HC-SR04
D_1	Motor driver	L9110S
K_1	Processing board	ESP32DEV
A_1	Current- and voltage-measuring board	MCU-219
L_1	LED panel, 256 LEDs	MAX7219
E_1 and E_2	Rechargeable battery	18650 3.7 V
F_1	Voltage converter	LM2596S

The compactness of the system was ensured as follows: the robots moved inside boxes, and the distance to the box walls was measured using ultrasonic sensors; when approaching the walls, the robot randomly changed its direction of movement.

As in the virtual agents, two types of resources were used in the tasks: LEDs on the LED panel (with specifying the number of LEDs lit) and the rotation speed of the robot's electric motors. Despite its simplicity, this scenario is common in real systems [4].

Due to technical limitations of the experimental study, it was impossible to implement differentiated energy consumption by each resource of a given type (e.g., use LEDs of different power); therefore, in-situ experiments were conducted only for scenarios with uniform consumption of energy by resources of one

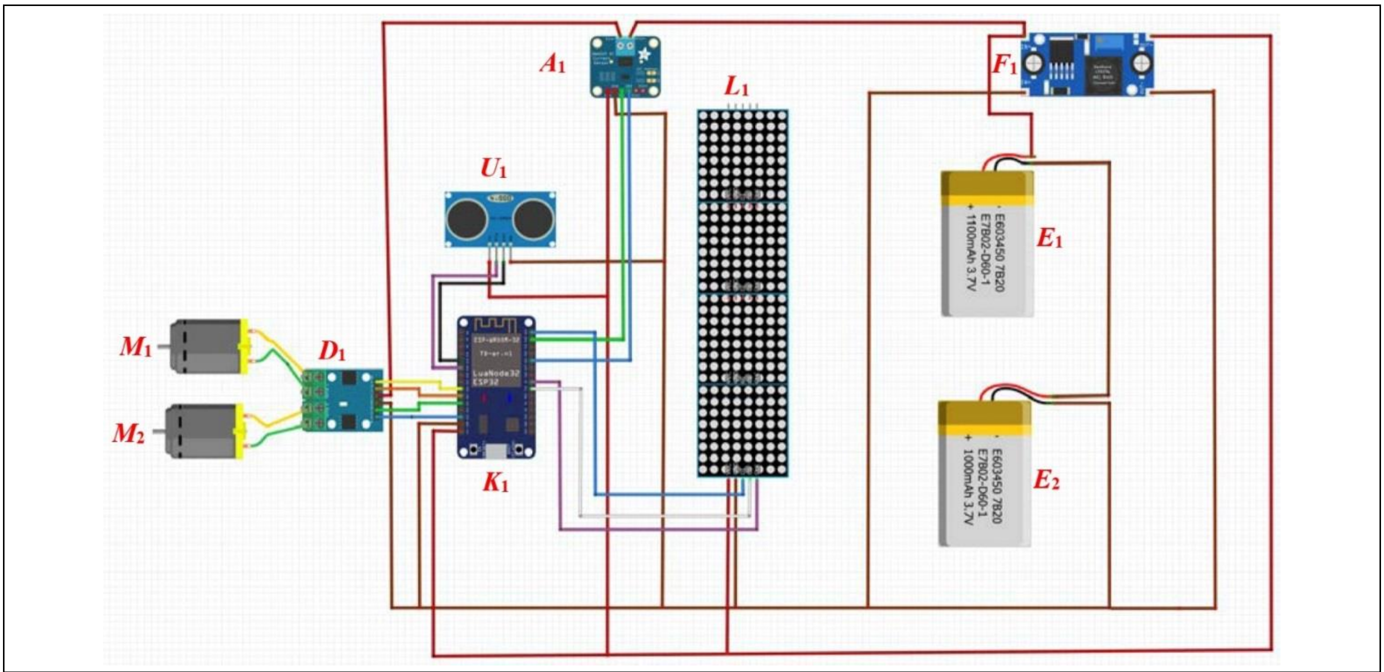

Fig. 6. The block diagram of a robot for the experimental mock-up of a DTS.

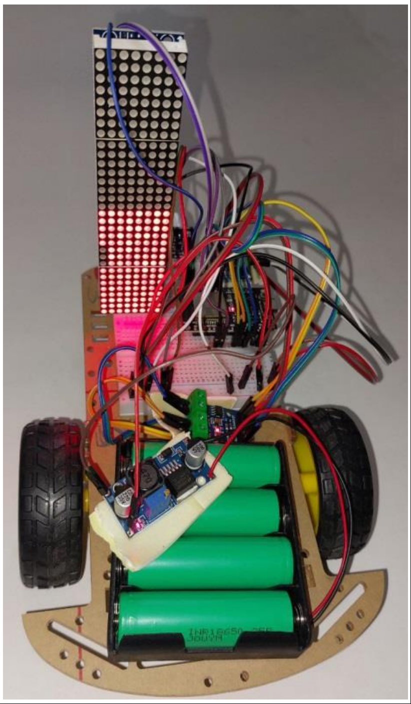

Fig. 7. A robot of the experimental mock-up: an enlarged photo.

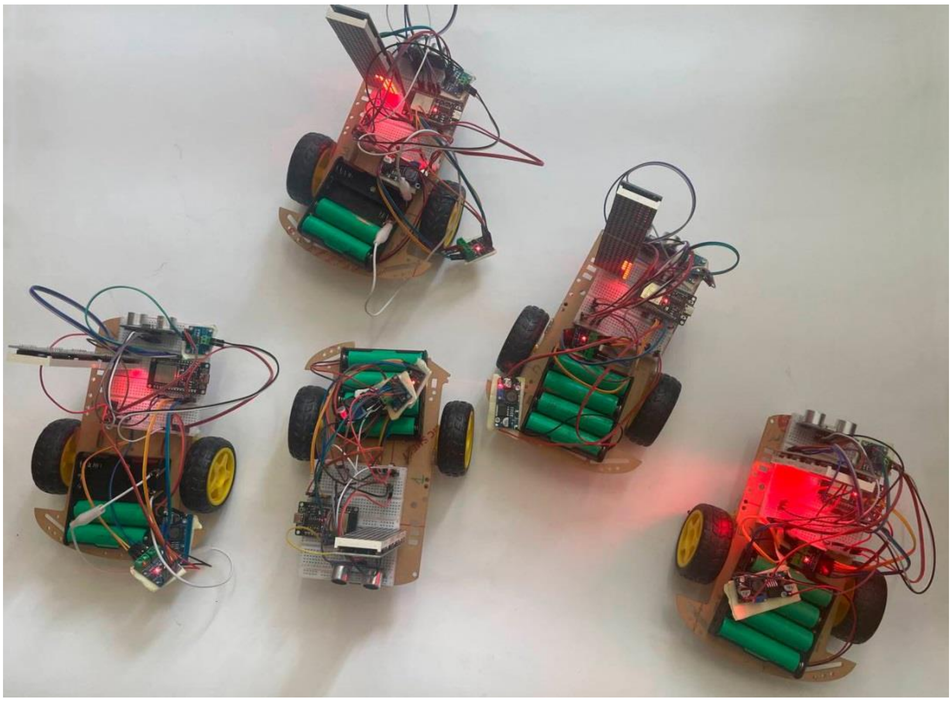

Fig. 8. The experimental mock-up with five robots.

type per unit of resource (similar to Examples 2.1and 2.2).

Before the experiments began, the specific energy consumption for both types of resources was measured, and the energy stored in the batteries was estimated. In particular, the assumption of a linear dependence of energy consumption on the consumed resource, adopted in most computer simulators [21], was checked. The energy consumption of each agent was measured using a special microchip. According to the measurement results, there is a rather high variance in specific energy consumption by agents for both types of resources, comparable to the absolute value of the measured quantity; however, this is acceptable for assessing the performance of the algorithm.

The results of the in-situ experimental study are provided in Figs. 9–12 below. In particular, Fig. 9 shows the average power of agents DTS1–DTS5 when loaded with several units of the first type of resource (the number of active LEDs). Similarly, Fig. 10 presents the average power of agents, over a long period of time, when using only the second type of resource (the rotation speed of electric motors). The confidence intervals correspond to level σ. To justify the form of the difference function (6) for the power consumption values measured, a linear regression was built, with the number of load units taken as the independent variable.

Note that, unlike the first type of resource (Fig. 9), "flattening" was observed for the second type of resource at high loads, and the function has a pronounced S-shape. However, for the comparability of the results with the virtual agents, the linear approximation was retained.

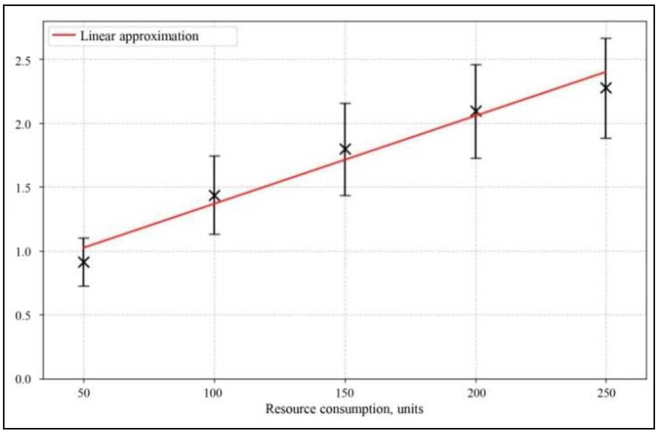

Fig. 9. The dependence of the average power consumption of agents on their load in resource units when fulfilling a task of type 1.

In the course of the experiments, the linear nature of energy consumption when fulfilling several tasks simultaneously was checked. Note that the linear nature of the dependence was confirmed.

Example 2.4. To assess the performance of the algorithm, control was applied to the DTS with real agents (robots) and, simultaneously, to that with virtual agents with the same parameters. Figures 11 and 12 show the energy dynamics of the system with virtual and real agents, respectively, at successive time instants.

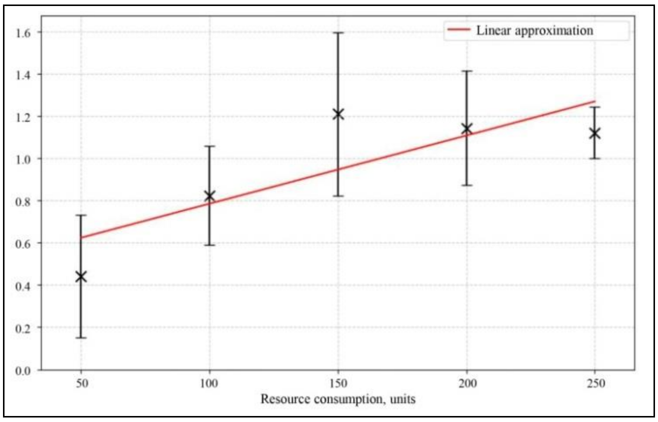

Fig. 10. The dependence of the average power consumption of agents on their load in resource units when fulfilling a task of type 2.

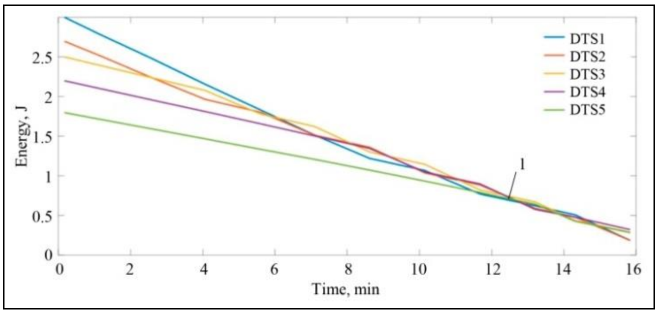

Fig. 11. The energy dynamics of system agents during operation (the DTS with virtual agents). The sampling step is 60 s.

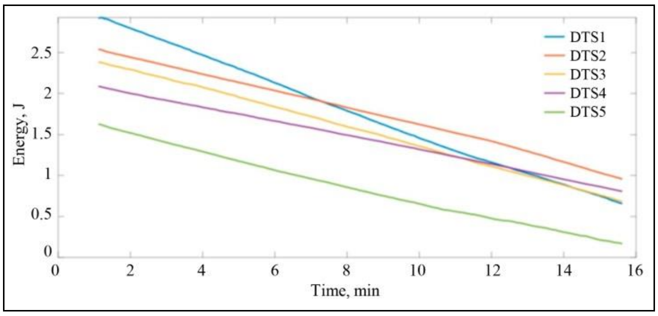

Fig. 12. The energy dynamics of system agents during operation (the DTS with real agents). The sampling step is 60 s.

According to these figures, in both cases, controlling task assignment among agents ensured a comparable lifetime (about 15 minutes) for the DTS.

In both cases, control was intended to equalize the energies of the agents. However, for the simulation model, control is more "adequate" in the sense of system behavior: at the junction point of the trajectories (point 1 in Fig. 11), the energies of all agents are reduced to one level, and the variance of the system energy is optimized. In a real system, this does not happen due to the nonlinearity in the energy consumption characteristics of the second type of resource (see Fig. 10), errors in determining the specific power, differences in the technical characteristics of agents, and other disturbances.

During testing on the mock-up, a series of similar experiments were conducted; they showed the repeatability of the results and the good performance of the algorithm under a significant spread of DTS parameters (battery capacity, energy consumption by motors, etc.).

3. A HEURISTIC METHOD FOR CHANGING THE SAMPLING STEP

The convergence and stability of dynamic programming algorithms are determined, in particular, by the choice of the sampling step at which the system is supposed to be stationary. If the system is described by a differential equation, this step can be calculated by specifying an estimate of the local error, e.g., using Euler's method. However, during experiments with the CP-SAT algorithm, we observed a feature of the algorithm that can be used to conclude on the stationarity of the system for a given sampling step; as a result, a heuristic approach can be applied to select the sampling step.

The heuristic approach consists in the following. For each sampling step, a time limit for finding a solution by the CP-SAT algorithm is specified; if the algorithm terminates much earlier than the time limit, the sampling step is increased; otherwise, it is reset to a certain default value. This approach is based on the following considerations: if the state of the system is stationary during the sampling step, then the values obtained at the previous time instant are a good approximation of the initial values for the system parameters optimized (task assignment among agents) at the next time instant, and the algorithm quickly converges to the optimal solution; otherwise, a lengthy search is required.

To verify the heuristic approach, the experiments from Examples 2.3and 2.4 (conducted previously with a fixed sampling step) were repeated using the algorithm with a dynamically changing step as described above.

Figure 13 shows the energy consumption for the DTS with virtual agents (see Example 2.3) with a dynamically changing sampling step.

Clearly, the sampling step was small at the start of the system and after point 1 (the junction of the trajectories). Compared to the uniform step (see Fig. 5), the number of time instants (the number of interventions in the behavior of the system) decreased by almost 80%.

The operation of the algorithm for the system from Example 2.4showed a similar picture, both for virtual (Fig. 14) and real (Fig. 15) agents.

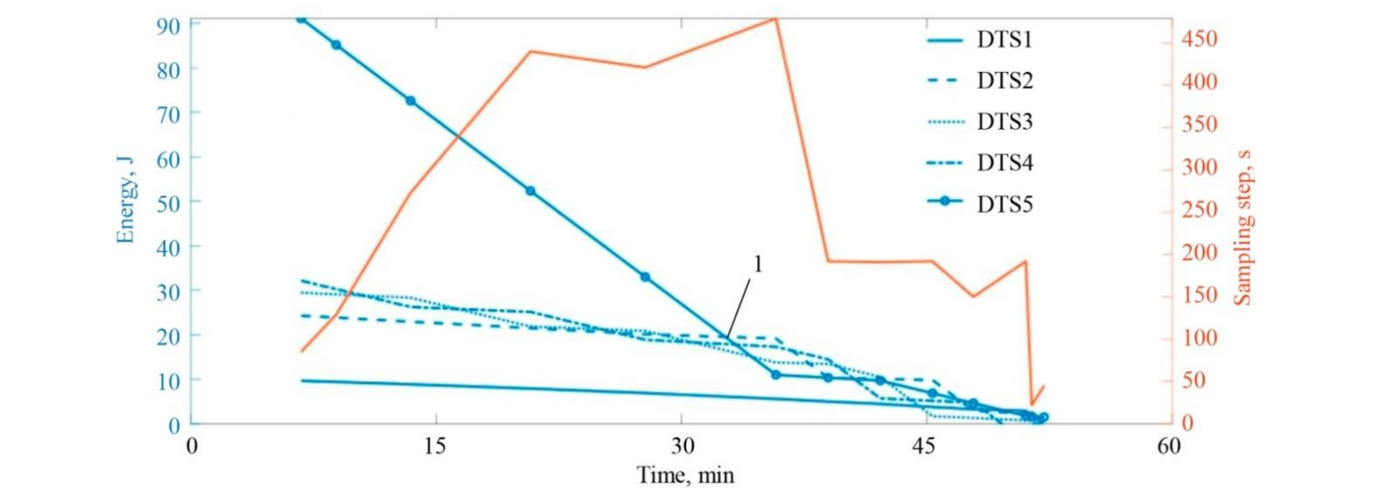

Fig. 13. The energy dynamics of virtual agents during operation. The sampling step is T = [30, 480] (s). DTS1–DTS5 are the energy reserve curves of the corresponding agents, and the orange line shows the dynamic sampling step.

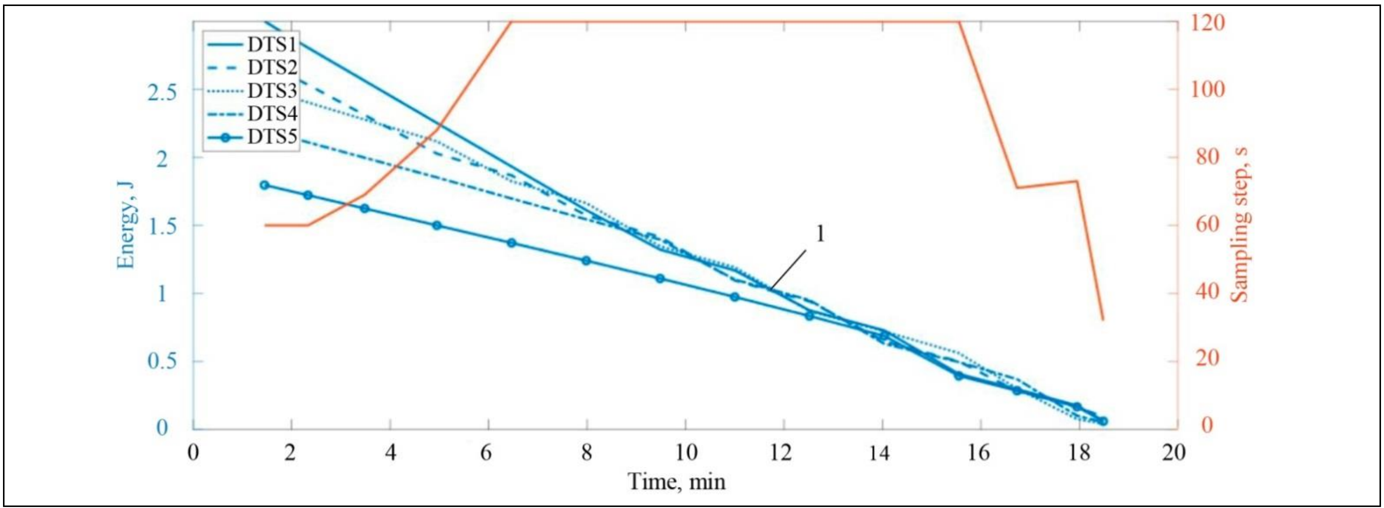

Fig. 14. The energy dynamics of real agents during operation. The sampling step is T = [60, 180] (s). DTS1–DTS5 are the energy reserve curves of the corresponding agents, and the orange line shows the dynamic sampling step.

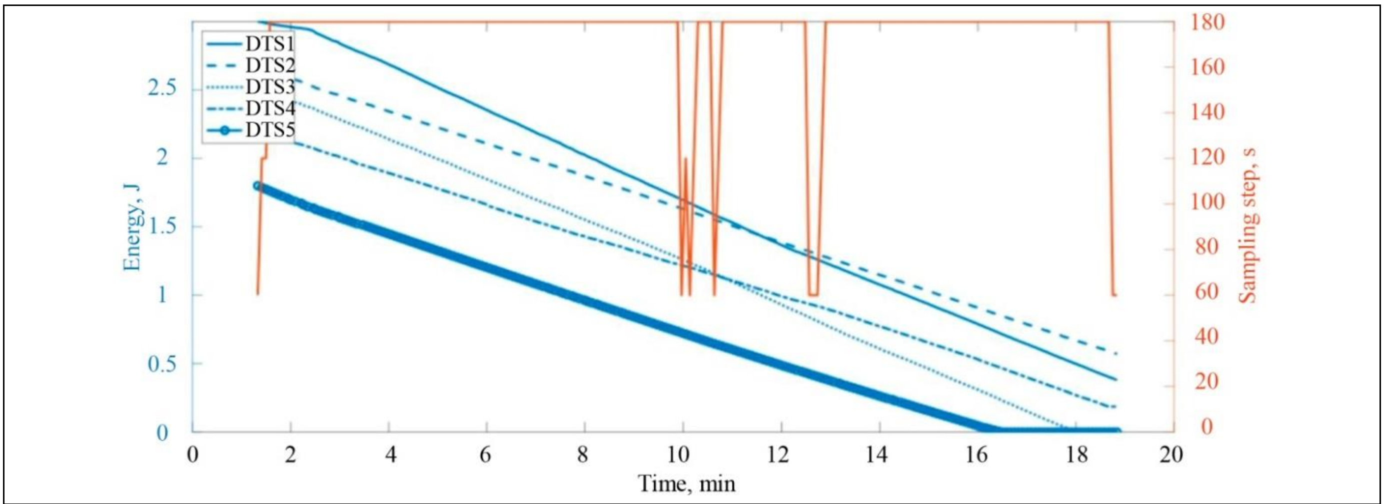

Fig. 15. The energy dynamics of real agents during operation. The sampling step is T = [60, 180] (s). DTS1–DTS5 are the energy reserve curves of the corresponding agents, and the orange line shows the dynamic sampling step.

A comparison of Figs. 11 and 14, as well as Figs. 12 and 15, shows that the main control quality parameter—the system lifetime—differs little between them, but the number of control instants in the second case is almost 1.5 times smaller.

CONCLUSIONS AND DISCUSSION

This paper has considered the practical problem of the optimal assignment of tasks among UVs in an enterprise's intelligent DTS environment in terms of energy consumption to ensure the availability of this system and increase the utilization factor of individual UVs. It is important both for providing the functional safety of DTSs and for improving the economic efficiency of the entire enterprise.

To solve this problem, an optimization-based mathematical model has been developed and implemented, including a special objective function and a set of constraints on the parameters of the system and individual UVs. The model assigns tasks among UVs with the necessary resources and ensures uniform energy consumption by individual agents. The optimization problem has been solved using dynamic programming methods with the application of the CP-SAT Boolean Satisfiability algorithm with integer constraints [24]. This algorithm has proven effective in solving such problems if approximate solutions can be accepted.

The performance of the model has been assessed both on virtual agents (computer simulations) and on an experimental mock-up representing a group of wheeled robots (see Fig. 8). Experience shows that the current low cost and availability of components make it possible to quickly create experimental mock-ups in laboratory conditions, and we believe that testing models on such mock-ups will become widespread. Testing on an experimental mock-up is extremely important since the characteristics of real systems would hardly be taken into account during computer simulations: all UVs, even from the same batch, differ, and their characteristics will vary over time; the characteristic curves for UV parameters included in the model may differ from the real ones, etc.

Experiments on a real system have confirmed the effectiveness of control even under a significant variation in the parameters of same-type agents in the DTS. The controlled energy dynamics of UVs in the real system (see Fig. 12) have not converged to a single trajectory, unlike the computer model (see Fig. 11). This is probably due to two features of real UVs. First, although UVs are of the same type, the specific power per unit of resource varies considerably between them, even within the same batch. Second, the model assumption on the linearity of energy consumption depending on the resource value is not always true (see Figs. 9 and 10), and additional calibration of this dependence is required.

While working with the optimization algorithm, we have discovered the feasibility of tuning the sampling step depending on the algorithm's time to find an acceptable solution under all constraints: a reduction in the running time of the optimization program indicates the possibility of increasing the sampling step. This heuristic technique requires further research, as it depends on the system type and the choice of initial values for the algorithm. However, it seems noteworthy: in addition to reducing computational cost, this technique significantly restricts the control center's intervention in the system's behavior.

On a modern average-speed computer, the model has demonstrated good performance with a sampling step (control cycle) of about 15 minutes for a system with 10^4 agents and 10^5 tasks, which is more than sufficient for an enterprise-level DTS.

A possible direction for further research is to develop adaptive self-tuning algorithms, e.g., a dynamically changing set of components of the objective function or constraints based on the DTS state data, in order to improve the efficiency of control.

Optimizing the distribution of DTS energy resources can also be part of the general safety monitoring problem of DTSs [31]. The optimal distribution of energy resources ceases to be a "black box" for the safety system, providing it with valuable information about the current state and its closeness to critical conditions. As a result, the safety monitor can pass from reactive to predictive safety analysis. In turn, the safety monitor can act as a meta-regulator that dynamically adapts admissible ranges in the energy resource optimization problem, ensuring that the system finds the most effective solution within the safe state space. Such an integrated approach is a prerequisite for creating complex and simultaneously reliable DTSs.

Acknowledgments. *This work was supported by the Russian Science Foundation, project no. 23-19-00338, https://rscf.ru/en/project/23-19-00338/.*

REFERENCES

1. *IEC 61508-4: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems - Part 4: Definitions and Abbreviations*, 2010.
2. *GOST* (State Standard) *R 27.102-2021: Reliability in Engineering. Object's Reliability. Terms and Definitions*, Moscow: Russian Standardization Institute, 2021. (In Russian.)

3. Promyslov, V.G., Availability Model Based on Network Calculus for Data Flow Processing System, *Large-Scale Systems Control*, 2024, no. 110, pp. 113–148. (In Russian.)
4. Härdtlein, C., Stadlbauer, H., and Daub, R., Energy-Oriented Optimization of the Task Allocation for Mobile Robots, *Procedia CIRP*, 2024, vol. 126, pp. 182–187.
5. Furnon, V. and Perron, L., *OR-Tools Routing Library*, Mountain View: Google Developers, 2024. URL: https://developers.google.com/optimization (Accessed June 18, 2025.)
6. Bukata, L., Sucha, P., Hanzalek, Z., and Burget, P., Energy Optimization of Robotic Cells, *IEEE Transactions on Industrial Informatics*, 2017, vol. 13, no. 1, pp. 92–102.
7. *Gurobi Optimizer Reference Manual*, Beverton: Gurobi Optimization, 2024. URL: https://docs.gurobi.com/projects/optimizer/en/current/index.htm1 (Accessed June 18, 2025.)
8. Krestovnikov, K.D. Mathematical Model and Control Algorithms for a Group of Ground Robots with Energy Resource Redistribution, *Information and Control Systems*, 2023, no. 6 (127), pp. 20–34. (In Russian.)
9. Kuhn, H.W., The Hungarian Method for the Assignment Problem, *Naval Research Logistics (NRL)*, 1955, vol. 52, pp. 83–97.
10. Carabin, G., Wehrle, E., and Vidoni, R., A Review on Energy-Saving Optimization Methods for Robotic and Automatic Systems, *Robotics*, 2017, vol. 6, no. 4, art. no. 39.
11. Chakraa, H., Guerin, F., Leclercq, E., and Lefebvre, D., Optimization Techniques for Multi-Robot Task Allocation Problems: Review on the State-of-the-Art, *Robotics and Autonomous Systems*, 2023, vol. 168, art. no. 104492.
12. Achterberg, T., Bixby, R., Gu, Z., et al., Presolve Reductions in Mixed Integer Programming, *INFORMS Journal on Computing*, 2020, vol. 32, no. 2, pp. 473–506.
13. Belousov, E.G., *Vvedenie v vypuklyi analiz i tselochislennoe programmirovanie* (Introduction to Convex Analysis and Integer Programming), Moscow: Moscow State University, 1977. (In Russian.)
14. Garey, M.R. and Johnson, D.S., *Computers and Intractability: a Guide to the Theory of NP-completeness*, New York: Freeman, 2009.
15. Kapustyan, S.G., Kalyaev, I.A., and Gaiduk, A.R., *Modeli i algoritmy kollektivnogo upravleniya v gruppakh robotov*, Moscow: Fizmatlit, 2009. (In Russian.)
16. Polyak, B.T., Khlebnikov, M.V., and Rapoport, L.B., *Matematicheskaya teoriya avtomaticheskogo upravleniya* (Mathematical Theory of Optimal Control), Moscow: LENAND, 2019. (In Russian.)
17. Bonabeau, E., *Swarm Intelligence: From Natural to Artificial Systems*, Cary: Oxford University Press, 1999.
18. Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, Cambridge: MIT Press, 2016.
19. Nesterov, Yu., *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, 2004.
20. *Handbook of Combinatorial Optimization*, Pardalos, P.M., Du, D.-Z., and Graham, R.L., Eds., New York: Springer, 2013.
21. Torabbeigi, M., Lim, G.J., and Kim, S.J., Drone Delivery Scheduling Optimization Considering Payload-Induced Battery Consumption Rates, *Journal of Intelligent & Robotic Systems*, 2020, vol. 97, no. 3–4, pp. 471–487.
22. Pozdnyakov, Yu.M., A Decomposition Scheme for Solving Problems of Integer-Valued Linear Programming, *USSR Comput. Math. Math. Phys.*, 1982, vol. 22, no. 1, pp. 58–69.
23. Nemhauser, G.L. and Wolsey, L.A., *Integer and Combinatorial Optimization*, New York: John Wiley & Sons, 1999.
24. Du, D. and Sun, J., *Advances in Optimization and Approximation*, Boston, MA: Springer, 1994.
25. Perron, L., Didier, F., and Gay, S., The CP-SAT-LP Solver, *Proceedings of the 29th International Conference on Principles and Practice of Constraint Programming (CP 2023)*, 2023, vol. 280, pp. 3:1–3:2.
26. Adams, W.P. and Forrester, R.J., Linear Forms of Nonlinear Expressions: New Insights on Old Ideas, *Operations Research Letters*, 2007, vol. 35, no. 4, pp. 510–518.
27. Forrester, R.J. and Waddell, L.A., Linearization Strategies for Binary Quadratic and Higher-Order Polynomial Programs, in *Encyclopedia of Optimization*, Pardalos, P.M. and Prokopyev, O.A., Eds., Cham: Springer, 2023, pp. 1–7.
28. Martínez-Clark, R., Cruz-Hernandez, C., Pliego-Gimenez, J., and Adrellano-Delgado, A., Control Algorithms for the Emergence of Self-organized Behaviours in Swarms of Differential-traction Wheeled Mobile Robots, *International Journal of Advanced Robotic Systems*, 2018, vol. 15, no. 6, art. no. 1729881418806435.
29. Ögren, P. and Sprague, C.I., Behavior Trees in Robot Control Systems, *Annual Review of Control, Robotics, and Autonomous Systems*, 2022, vol. 5, pp. 81–107.
30. Vasiliev, A. and Dalyaev, I., Simulation Method for the Transport System of a Small-Sized Reconfigurable Mobile Robot, *Machines*, 2021, vol. 9, no. 1, art. no. 8.
31. Jharko, E.Ph., Abdulova, E.A., Promyslov, V.G., and Semenkov, K.V., Methodology for Calculating Availability as a Function of the Safety Monitor of a Dynamic Technical System, *Journal of Information Technologies and Computing Systems*, 2024, no. 3, pp. 60–72. (In Russian.)

This paper was recommended for publication by R. V. Meshcheryakov, a member of the Editorial Board.

Received July 3, 2025,
and revised September 11, 2025.
Accepted October 1, 2025

Author information

Semenkov, Kirill Valer'evich. Cand. Sci. (Phys.–Math.), Trapeznikov Institute of Control Sciences, Russian Academy of Science, Moscow, Russia
✉ semenkov@ipu.ru
ORCID iD: https://orcid.org/0000-0003-0865-9072

Abdulova, Ekaterina Alekseevna. Researcher, Trapeznikov Institute of Control Sciences, Russian Academy of Science, Moscow, Russia
✉ consoft@ipu.ru
ORCID iD: https://orcid.org/0000-0002-7843-5202

Jharko, Elena Filippovna. Cand. Sci. (Phys.–Math.), Trapeznikov Institute of Control Sciences, Russian Academy of Science, Moscow, Russia
✉ zharko@ipu..ru
ORCID iD: https://orcid.org/0000-0002-8895-4786

Promyslov, Vitaly Georgievich. Cand. Sci. (Phys.–Math.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ vp@ipu.ru
ORCID iD: https://orcid.org/0000-0003-1919-8718

Cite this paper

Semenkov, K.V., Abdulova, E.A., Jharko, E.F., and Promyslov, V.G., Optimizing Task Assignment among Unmanned Vehicles in Terms of Energy Consumption. *Control Sciences* **6,** 69–83 (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