Автоматизовані системи управління та прилади автоматики. 2025. № 4 (187)

UDC 004.056.55:004.312.2 **DOI:** https://doi.org/10.30837/0135-1710.2025.187.063

S. Yenhalychev, S. Semenov

EXPERIMENTAL STUDIES OF THE TASK PLANNING METHOD IN DISTRIBUTED INFORMATION SYSTEMS TAKING INTO ACCOUNT METADATA UNCERTAINTY

The article is devoted to task planning in distributed information systems under conditions of uncertainty and incompleteness of metadata regarding the duration of operations, communication volumes, and resource availability. **The subject of the study** is the method of task planning in distributed information systems, taking into account the uncertainty of metadata. **The purpose of the article** is to experimentally investigate a method for planning tasks in distributed information systems with uncertain and incomplete metadata, which combines a hybrid graph model with a three-module artificial intelligence core and provides increased efficiency and robustness of planning while complying with SLO and RBAC constraints. **Research objectives:** to form a hybrid representation of workflows based on a combination of DAG structure and GERT model; to develop a planner architecture incorporating a GAT graph attention network, a PPO reinforcement learning agent, and a Bayesian Optimization module; to define an SLO-oriented reward function and constraint system, taking into account RBAC; to create a reproducible experimental test bed with controlled uncertainty injection; to perform comparative, ablation, and robustness analysis of the proposed method's productivity relative to the baseline FCFS and HEFT algorithms, as well as DAG-only and GERT-only variants. **Research methods:** hybrid graph modeling DAG+GERT, graph attention network GAT for forming a contextual state representation, PPO agent for making decisions on task placement, Bayesian optimization for tuning policy parameters, and statistical evaluation methods. **Main results.** The proposed method provides a significant reduction in the median task completion time (*makespan*) compared to HEFT, DAG-only, and GERT-only, an increase in the proportion of tasks performed within SLO, and a reduction in weighted average delay, demonstrates "tail compression" of the delay distribution, and maintains stability of results in the case of multiplicative noise of durations ±20 %, subject to distribution shifts and correlated parameter deviations. At the same time, planning overhead remains in the millisecond range and does not exceed acceptable limits for online scenarios.

Keywords: distributed information systems; task scheduling; directed acyclic graph; GERT networks; graph neural networks; reinforcement learning; Bayesian Optimization; service levels (SLO); access control (RBAC); robustness.

Introduction

In modern distributed information systems, it is increasingly necessary to plan large sets of dependent tasks under conditions of variable load, resource heterogeneity, and incomplete input metadata. For such systems, it is critical to simultaneously minimize the completion time of a set of tasks (makespan), ensure compliance with service levels (SLO – service level objective), and take into account role-based access control (RBAC) policies.

Classic planning approaches focus on directed acyclic graphs (DAG), in which vertices correspond to tasks and edges correspond to dependencies and data exchanges. A broad class of heuristic algorithms is based on this model, including HEFT (Heterogeneous Earliest Finish Time) and its numerous variants, which demonstrate high-quality scheduling in heterogeneous computing environments with deterministic estimates of task and communication durations.

However, in real distributed systems, the duration of operations, the bandwidth of communication channels, and even the structures of workflows themselves change under the influence of external factors, and the available metadata is noisy, partially outdated, or incomplete. This requires robust scheduling methods capable of maintaining acceptable schedule quality under stochastic and adversarial parameter perturbations. The robustness of graph schedules is being actively researched, but the vast majority of work is either limited to static stochastic duration models or focuses on optimizing a single criterion, primarily makespan.

An additional complexity is the need to model not only sequential dependencies, but also probabilistic branches, repetitions, conditional transitions, and failures inherent in real-world event processing and telemetry scenarios. For such tasks, it is natural to use GERT (Graphical Evaluation and Review Technique) networks, which allow the inclusion of branch passage probabilities and duration distributions in the model, but historically have been used primarily in project management rather than in operational planning tasks in information systems.

In recent years, intelligent planning methods using graph neural networks and reinforcement learning have been actively developed to optimize scheduling in complex environments. A number of approaches have been proposed in which a reinforcement learning agent makes decisions about the placement of tasks in a DAG, and graph networks (GCN, GNN) or graph attention (GAT – Graph Attention Network) are used to construct a topology-sensitive state. Despite significant progress, most of these works either do not take into account the probabilistic nature of branches or do not integrate SLO requirements and access policies into a single model.

In this work, these problems are solved by combining a hybrid DAG+GERT graph model with a three-module planner core based on artificial intelligence methods, which allows us to explicitly describe the uncertainty in the execution structure and task parameters, and to train an adaptive planning policy focused on SLO and access restrictions.

Literature review

The issue of task scheduling in parallel and distributed systems has traditionally been considered within the framework of deterministic models, where task durations and exchanges are fixed or accurately estimated. Classic monographs and reviews [1, 2] focus on building effective heuristics for DAG representations, including clustering, leaf scheduling, and task duplication methods.engr.colostate.edu One of the most cited algorithms is HEFT (Heterogeneous Earliest Finish Time) [3, 4], which provides a good compromise between schedule quality and computational complexity for heterogeneous environments. Based on HEFT, a large number of modifications (E-HEFT, Rob-HEFT, PDHEFT, etc.) [5, 6] have been proposed that take into account load balancing, power consumption, or specific hardware constraints.

However, deterministic models poorly reflect the dynamics of modern distributed systems, where task durations and network delays depend on current load, queues, and external factors.

This has led to the development of robust planning [7, 8], which considers stochastic or interval durations, and the quality of the schedule is evaluated not only by the expected makespan, but also by the distribution of this indicator, slack, and sensitivity to disturbances. A number of robustness metrics and corresponding heuristics have been proposed, particularly for network and cluster environments, but most approaches remain static and do not use online policy updates based on telemetry [9].

A separate class of works [10, 11] is devoted to the application of deep reinforcement learning (DRL) to DAG planning problems in cloud and cluster systems. In these works, the DRL agent learns on a simulated environment or real telemetry to minimize makespan, energy consumption, or combined criteria.

A number of studies [12, 13] demonstrate the advantages of DRL approaches over classical heuristics under variable load conditions, but most of them rely on pure DAG models and do not take into account probabilistic branches, failures, or cyclic fragments of workflows.

In parallel, graph neural networks (GNN) [14, 15] are being developed for planning and optimization tasks. Schemes are proposed in which GCN or GAT modules build representations of task graphs (job shop, workflow, access network), and these representations are then used either as a heuristic function [16] or as a policy component [17]. It has been shown that GNN approaches scale better with respect to graph size and topology complexity than classical tabular or manually designed features. However, most works work with deterministic DAG structures and do not use the combined power of stochastic networks such as GERT to model uncertainty.

GERT-type networks [18] were historically developed for analyzing stochastic networks in project management and operations research problems. They allow modeling the probability of branch traversal, as well as the distribution of time or other additive attributes on edges, making them a natural tool for describing complex technological schemes with branches and cycles [19]. Despite this, the use of GERT in the context of task planning in distributed information systems remains limited, and the combination of GERT with modern GNN/DRL methods has hardly been explored.

Another important component of real-world systems is access policies, which determine which tasks can be performed on which resources. The most commonly used formal model here is role-based access control (RBAC), standardized by NIST and widely implemented in corporate environments [20]. In parallel with this, the area of service reliability engineering has developed an established practice of operating with service indicators and service level objectives (SLOs), which serve as the basis for making decisions about operational trade-offs [21]. Existing works on DRL planning largely lack the integration of SLO-oriented metrics and RBAC constraints into a single learning framework.

Thus, analysis of the literature reveals the following gaps:

- Insufficient use of stochastic networks such as GERT to model uncertainty in the structure and parameters of workflows.

- Limited integration of graph neural networks and DRL agents with GERT representation in planning tasks.

- Almost complete absence of works in which the planning policy is simultaneously optimized for makespan, SLO-oriented indicators, and compliance with RBAC constraints.

The purpose of the article is to experimentally investigate a method for planning tasks in distributed information systems with uncertain and incomplete metadata, which combines a hybrid graph model with a three-module artificial intelligence core (GAT, PPO, Bayesian optimization) and provides increased efficiency and robustness of planning while complying with SLO and RBAC constraints.

To achieve this goal, the following main tasks are solved.

1. Form a hybrid model of work processes based on a combination of a DAG structure with a GERT network, which allows simultaneously describing deterministic dependencies, probabilistic branches, repetitions, and failures.

2. Develop a scheduler architecture in which a graph attention network (GAT) builds a representation of the system state, a deep reinforcement learning agent (PPO) makes decisions about task placement, and a Bayesian optimization module performs automated tuning of hyperparameters and threshold criteria.

3. Formulate an SLO-oriented reward function and constraint system that simultaneously considers makespan, deadline probability, and compliance with RBAC policies.

4. Build a reproducible experimental testbed with controlled load scenarios, metadata uncertainty models, and sets of baseline algorithms (FCFS, HEFT, simplified GERT/DAG modes) for comparison.

Conduct a comprehensive experimental analysis, including assessment of robustness to parameter perturbations, ablation analysis of the contribution of individual modules (GAT, PPO, BO), and statistically sound comparison with baseline approaches.

Main part

1. Formulation of research hypotheses, experimental design, and performance indicator system

Let us begin with some terminological explanations. In this study, metadata is understood as all parameters that describe the behavior of tasks and operations in the DAG+GERT model and are transmitted to the scheduler as input information. Formally, the set of metadata is defined by:

$$\mathrm{M} = \left\{ \tau_i, \sigma_i^2, c_{ij}, p_{ij}, r_i, D_i, w_i, RBAC_i \right\},$$

where: τ_i – task duration estimation;

σ_i^2 – dispersion or other statistical characteristics of duration;

c_{ij} – cost or delay of data transfer between tasks $i \to j$;

p_{ij} – probability of transition in the GERT branch;

r_i – task requirements for resources;

D_i – local deadline;

w_i – weight coefficient of the task for calculating the delay;

$RBAC_i$ – access policies and restrictions on task execution.

Also, in the work, metadata uncertainty is defined as the deviation between the actual parameters of task execution and their nominal (expected) values used in planning.

Formally, uncertainty is specified by the perturbation operator:

$$\mathsf{M}' = \mathsf{M} \circ (1+\Delta),$$

where Δ – a random or adversarial value that models errors in the input parameters.

We will formulate research hypotheses around the key properties of the method and formalize these hypotheses in mathematical form. Let M be the makespan, R be the proportion of tasks completed within the target SLOs, L be the weighted average delay, Θ be the overhead costs of planning, and S be the scalability (acceleration) index. Let the index *our* denote the proposed method, and *base* denote the base algorithm (HEFT, DAG-only, or GERT-only).

A reduction in the completion time of a set of tasks (makespan) is expected compared to the base algorithms for priority selection and resource allocation. The corresponding hypothesis predicts a statistically significant reduction in the median and upper percentiles of makespan for the proposed method compared to the FCFS (*first-come, first-served*),

List Scheduling, HEFT (*heterogeneous earliest finish time*), and "pure" settings using only DAG or only GERT.

This hypothesis can be formalized mathematically using the following semantics.

$$median(M_{our}) < median(M_{base}),$$

$$F_{our}^{-1}(0.95) < F_{base}^{-1}(0.95).$$

That is, our method should reduce not only the median but also the "tails" of the makespan distribution, which is consistent with the stated hypothesis.

1. We expect to see an increase in deadline reliability, i.e., an increase in the proportion of tasks completed within the specified SLOs under the same resource constraints; this hypothesis is evaluated both at the aggregate level and in terms of task subclasses.

$$R_{our} = \frac{1}{N}\sum_{i=1}^{N} 1(C_i \le D_i) > R_{base}.$$

2. It is assumed that the average decision-making time by the planner (planning overhead) remains acceptable for online scenarios and does not conflict with the delay requirements in queue control loops.

$$\Theta_{our} = \frac{1}{K}\sum_{k=1}^{K} t_k^{plan} \le \Theta_{max}.$$

3. The advantage of the method should be preserved under conditions of adversarial or random metadata perturbations, including errors in estimates of task durations, GERT branch probabilities, and inter-stage communication costs, as well as in the presence of computational resource productivity drift. We formalize this hypothesis as follows.

Let the noise injection operator:

$$\tau_i' = \tau_i(1+\xi_i), \xi_i \sim U[-\alpha, \alpha].$$

Then robustness is formulated as:

$$\Delta M(\alpha) = \frac{|M(\alpha) - M(0)|}{M(0)} \le \varepsilon,$$

$$R(\alpha) \approx R(0),$$

$$F_{our}^{-1}(0.95, \alpha) < F_{base}^{-1}(0.95, \alpha).$$

4. Predictable scalability is expected. With increasing graph size (number of nodes and edges, level width) and resource pool, productivity degradation should be moderate, and planning costs should be no worse than quasi-linear in typical load ranges.

Let's define:

$$S(N) = \frac{M(1)}{M(N)}.$$

In this case, the hypothesis takes the form:

$$S_{our}(N) \ge S_{base}(N),$$

$$\frac{d\Theta}{dN} \approx 0.$$

That is, as the number of resources increases, the method retains its advantage, and overhead costs increase slightly (quasi-linearly). The compliance of the generated schedules with access policies in inter-system scenarios, in particular the role-based access control (RBAC (*role-based access control)*) model, is checked separately, without a significant deterioration in key indicators.

The experimental design combines synthetic and real-world scenarios. Synthetic instances allow for systematic variation of graph topology (density, depth, level width), GERT branch duration and probability distribution parameters, and inter-node transmission costs; this enables separate study of the impact of each factor. Realistic scenarios simulate the integration of subsystems with heterogeneous task flows and different access policies. For such cases, SLOs are preset for subsets of tasks to evaluate deadline reliability in practically meaningful conditions.

Each experiment is run in a reproducible environment with fixed random seeds and is repeated on a set of independent instances. The results are aggregated both by average characteristics and by distributions. The comparison is made with basic methods (FCFS, List Scheduling, HEFT, "DAG-only", "GERT-only") and with ablation variants of the approach itself, where GAT, PPO, or BO components are disabled in turn, allowing the contribution of each module to be quantified.

Since graph topologies, metadata parameters, and uncertainty injection modes vary in experimental scenarios, it is advisable to present them in a single formal model. Let each input instance of a workflow be described by a directed graph

$$G = (V, E),$$

where for each task $i \in V_i$, the nominal duration τ_i, local deadline D_i, weight coefficient w_i, resource requirements, and, if there are GERT branches, transition probabilities p_{ij} are specified. Communication costs between tasks are described by values c_{ij}.

To model uncertainty, graph parameters undergo controlled changes. Task durations vary according to a multiplicative model:

$$\tau_i^{'} = \tau_i(1+\xi_i),\ \xi_i \sim U[-\alpha,\alpha],$$

which corresponds to situations where estimates come from incomplete or noisy metadata.

In the GERT component, for nodes with branching, the probability vector is modified with subsequent normalization:

$$p_{ij}^{'} = \frac{p_{ij}(1+\eta_{ij})}{\sum_{k\in\Gamma(i)} p_{ik}(1+\eta_{ik})},\ \eta_{ij} \sim U[-\beta,\beta],$$

which reproduces the instability of work flow statistics.

The cost of communications is disturbed lognormally:

$$c_{ij}^{'} = c_{ij} exp(\zeta_{ij}),\ \zeta_{ij} \sim LN(0,\sigma^2),$$

which is consistent with empirical delay profiles in heterogeneous computing environments.

To reproduce correlated parameter deviations within graph levels, a Gaussian copula is used:

$$(\xi_1,\ldots,\xi_m) \sim Copula_\rho,\ \rho = 0.3.$$

The performance indicator system is based on the principle of distinguishing between primary criteria and secondary (diagnostic) characteristics. Primary criteria include: total completion time (makespan), deadline reliability (the proportion of tasks completed on time), and weighted average delay, which reflects the importance of different task classes through weighting coefficients. Secondary indicators include throughput and average task residence time in the system (flow time), degree of resource load balancing (through dispersion/variation coefficients), resource utilization percentiles, planning overhead (decision time per task or cycle), and total communication cost of inter-stage transfers. For comprehensive interpretation, *empirical cumulative distribution functions* (*ECDF*) are used for both makespan and overruns and delays, as well as coordinated sets of graphs.

Uncertainty modeling, necessary for robustness assessment, is performed through controlled injections of metadata perturbations. In particular, multiplicative noise of durations in characteristic ranges (± 10–30 %), perturbations of GERT branch probabilities with normalization, drift of resource productivity over time, and variations in communication costs are used. Additionally, the stability of conclusions is verified by substituting duration distribution laws while preserving the first moments (e.g., Gamma ↔ Weibull) and in the presence of correlations between tasks within the same graph level. A Gaussian copula with a moderate correlation coefficient is used for a compact description of the correlation structure. Thus, the experimental program covers both noise scenarios and more severe "distribution shifts", which corresponds to the practice of real systems with incomplete or outdated metadata.

The statistical methodology involves the use of non-parametric criteria for paired and unpaired comparisons (Wilcoxon, Mann-Whitney criteria) in combination with effect size estimation (Cliff's δ) and *confidence interval (CI)* construction using the bootstrap method with a sufficient number of replications. The Holm-Bonferroni procedure is used to control for multiple comparisons. The significance level is set at 0.05 with the reporting of exact p-values and corresponding δ. All experiments are performed in a "fixed environment" with complete fixation of library versions and data generation scripts. Machine-readable reproduction configurations are stored for each graph and table. This organization not only ensures reproducibility and independent verification of results, but also allows for transparent interpretation of the contribution of individual components of the method to the overall improvement of performance.

2. Architecture of the experiment planner prototype

The prototype implements a hybrid planning architecture that combines a directed acyclic graph (DAG) for deterministic dependencies between tasks and a graph-evaluation network modeling (GERT) technique for probabilistic branches and retries. Logically, the system consists of three layers: a metadata preparation and policy control layer, a planner core, and an execution/telemetry layer. Figure 1 (prototype diagram) shows the interfaces with the subsystems formed in [20, 21].

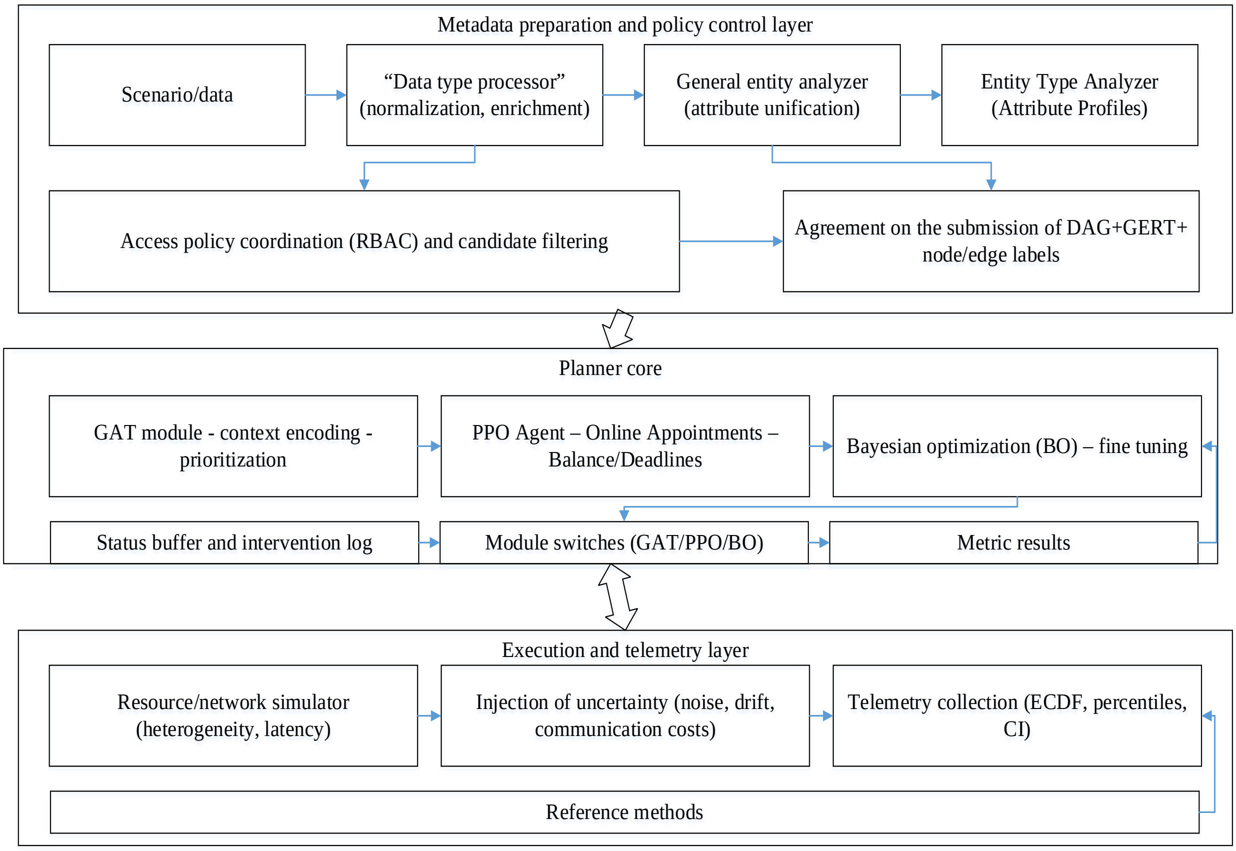

Fig. 1. Architecture of the prototype experiment design

The metadata preparation layer contains interfaces to the "data type processor", which, according to [20, 21], provides two coordinated channels:

– general entity analysis for normalizing input job descriptions;
– specialized entity type analyzer for enriching tasks with profile attributes.

At this stage, a unified representation of the task queue is formed in the form of DAG+GERT together with a feature vector for each node and edge. Node features include (with abbreviations expanded on first use):

– a priori duration estimates (mean and variance);
– level/path identifiers;
– input/output degrees;
– resource requirements;
– local deadlines;
– access policy labels (RBAC (role-based access control)).

Edge features include GERT branch probabilities, expected exchange volumes, and transmission costs between stages. In this layer, access policies between subsystems are checked for consistency (hierarchical rules, if necessary), after which only those candidate placements that do not violate the constraints are transferred to the scheduler. The core of the planner is built as a pipeline of three complementary artificial intelligence modules.

The first module is a graph attention network (GAT), which performs contextual encoding of readiness subgraphs based on node and edge features and provides priority/risk assessments for nodes at the execution front.

Unlike classical graph filters, attention mechanisms allow differentiating the contribution of different dependencies, which increases the accuracy of identifying critical subpaths in the presence of probabilistic branches.

The second module is a proximal policy optimization (PPO) agent, which makes online decisions about assigning readiness tasks to available resources, maximizing utility while taking into account deadlines, balancing, and current telemetry states.

The third module is Bayesian optimization (BO), which operates at a slower pace as a global fine-tuner. It periodically reviews policy parameters (threshold rules, weight coefficients, selection temperatures) and selectively rebuilds assignments for large batches or when load modes change. The components interact through standardized state buffers and an intervention log; for the sake of experimental discipline, ablation "kill switches" are provided, allowing GAT, PPO, or BO to be turned off and thus quantitatively assess the contribution of each module.

The execution and telemetry layer implements a reproducible test bed. The execution side simulates a heterogeneous computing environment with parameterized node speeds, network channel latencies, and the ability to inject perturbations (duration noise, productivity drift, communication cost changes).

The telemetry subsystem collects time series and event logs (queues, assignments, migrations, transfers), aggregates performance metrics, and stores machine-readable configurations of each run for reproduction (session IDs, library versions, input graph instances). Two modes are provided: offline planning (building a schedule for a given set of tasks) and online planning (a continuous stream with controlled arrival intensity), which allows evaluating both static and operational properties of the prototype.

Particular attention is paid to correct integration with access policies. Before each assignment step, candidates are filtered according to roles and contextual rules (RBAC), with verification performed before the PPO agent makes a decision to avoid unjustified "rollbacks" due to violations of restrictions. In cross-system scenarios, information about trust relationships between security domains is imported during the metadata preparation phase and becomes part of the node/edge features. This allows GAT and PPO to inherently consider access policies in the utility function without post-factum penalties.

Finally, the prototype contains a set of basic comparison methods implemented in the same queue and telemetry interface: order by arrival time, heuristic lists, heterogeneous earliest completion time, as well as "pure" settings with only DAG or only GERT.

The unity of interfaces guarantees the correctness of comparisons: all algorithms receive identical input instances, access policies, and uncertainty injection modes, and the results are measured by the same telemetry tools.

This architecture and test bench design ensures transparent mapping from hypotheses and indicators to experimental results, which will be presented in the following sections.

3. Comparison methods, ablation analysis, and training protocol

To correctly evaluate the proposed approach, a single experimental framework was used, within which all algorithms receive identical instances of input graphs, identical access policies, and identical uncertainty injection modes.

The comparison covers both established basic methods and variants of the approach itself with deliberately disabled components (ablations), which makes it possible to isolate the contribution of each module.

The following methods serve as the main elements of prototyping.

1. FCFS (*first-come, first-served*). Placing tasks in order of arrival when resources are available.

2. List Scheduling. Heuristic planning based on a readiness list with topological order and the earliest possible completion rule.

3. HEFT (*heterogeneous earliest finish time*). Ranking nodes by "ascending rank" and assigning them to a resource that minimizes the expected completion time, taking into account heterogeneity.

4. "DAG-only". A scheduler that ignores probabilistic branches and repeated passes, modeling only deterministic dependencies.

5. GERT-only. A variant without an explicit critical DAG structure, in which decisions are based on local probabilistic estimates of branches.

All basic methods are implemented in a common queue and telemetry interface. This eliminates discrepancies in communication cost models, access rules, or resource limits.

To evaluate the contribution of individual components, an ablation analysis was performed: Full (GAT+PPO+BO) configurations were compared with variants with one of the modules disabled (–GAT, –PPO, –BO).

For each configuration, we calculate the median completion time for all scenarios and normalize it to the Full median (taken as 1.0).

Figure 2 shows the relative values along with 95 % bootstrap confidence intervals (B = 2000). Values less than 1.0 indicate improvement, while values greater than 1.0 indicate degradation relative to Full. Statistical conclusions are made according to the bootstrap-CI protocol. At the same time, checks are based on Cliff's δ and permutation tests.

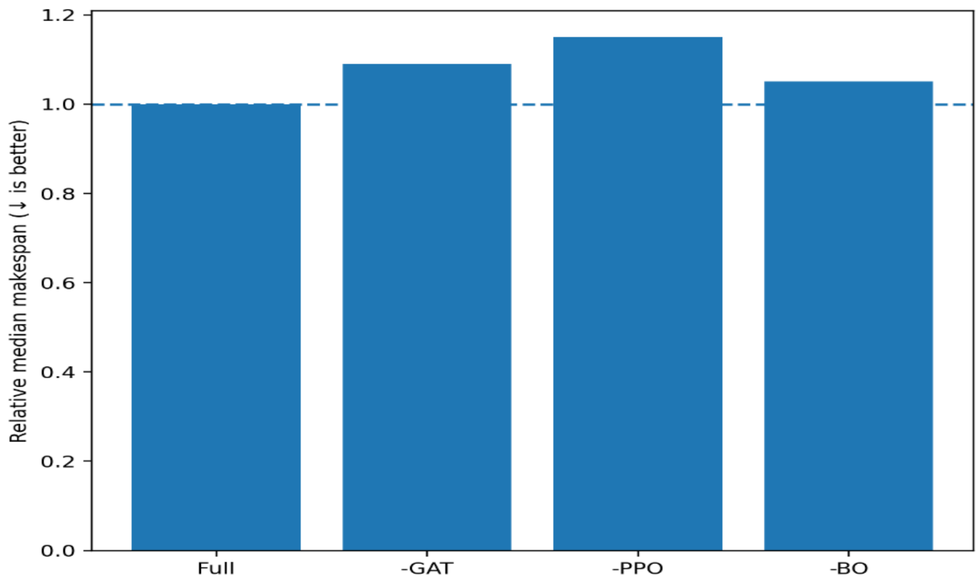

Fig. 2. Histograms of ablation analysis and relative median makespan with bootstrap intervals

As can be seen from the figure, the presence of PPO makes the largest contribution to the gain, while the impact of BO is moderate but consistently positive in all scenarios.

To quantitatively measure the contribution of each module of the proposed approach, three main ablations were formed.

1. "GAT". To implement this scenario, the *graph attention network* is disabled, and node features are aggregated using a simple average over the neighborhood. In addition, the prioritization of the execution front is set by static indicators—levelness and local slack.

2. "PPO". To implement this scenario, *proximal policy optimization* is disabled. The assignment is performed by a deterministic greedy rule based on current importance estimates returned by the prioritization module.

3. "BO". To implement this scenario, *Bayesian optimization* is disabled. Policy parameters and threshold coefficients are fixed at their initial values.

Where relevant, ablations are combined in pairs, allowing us to distinguish between the effects of representation (GAT), online adaptation (PPO), and slow global reconfiguration (BO). All modes use the same invalid-action masking. Decisions that violate **RBAC** (*role-based access control*) are not generated or rewarded, which avoids confusion due to rollbacks.

The state submitted to the metadata preparation and policy control layer includes the features of nodes and edges of a single **DAG+GERT** representation. These are predicted averages, duration variances, node "criticality" (GAT rating), slack relative to the local deadline, input/output degrees, resource queue delays, estimated data transfer costs, and aggregated resource utilization metrics. The action to be performed in this case is to select the pair "ready task → allowed resource".

Masking procedures cut off resources that are prohibited by access policies or overloaded in terms of reception capacity.

The reward function is formulated in increments (per-step). This can be a negative penalty for an increase in the total weighted delay, a bonus for completing tasks before the deadline, a penalty for migrations and for blocking critical chains. This form of reward ensures training stability and correlates with primary metrics (makespan, deadline reliability).

Training takes place in two phases. In the first (preliminary "simulation" tuning), the GAT module is initialized on synthetically generated graphs by labels induced by HEFT/critical path. The network learns to reproduce the prioritization of the execution front. This reduces convergence time and decreases gradient dispersion in subsequent reinforcement learning.

In the second phase, reinforcement learning is performed using **PPO**. A discount factor of $\gamma \approx 0.985$ is used. Advantages are evaluated using *generalized advantage estimation* ($\lambda \approx 0.95$). Policy updates are pruned with a parameter $\varepsilon \approx 0.2$, with low entropy regularization (to avoid early policy "freezing").

Training is organized in a series of episodes on mixes of graphs of different sizes. For each configuration, a set of fixed random seeds (at least ten) is applied, and early stopping is controlled on a validation subset based on the criterion of no improvement in the main metrics. **BO** (*Bayesian optimization*) runs asynchronously in a slower cycle. Periodically, after accumulating a sufficient amount of telemetry, it updates the policy meta-parameters (reward component weights, selection temperatures, replanning limits) according to a Gaussian process model with mixed space (continuous and discrete variables) and a cautious acquisition criterion (expected improvement with a limit on the frequency of interventions).

The following strategies are used to avoid overfitting.

– Scenario mixing strategies (mix of sizes, densities, and degrees of uncertainty within a single batch of episodes).

– Domain randomization of communication cost parameters.

– "Frozen" validation graphs on which "fine" tuning is not performed.

All results are presented as averages across repetitions with 95 % confidence intervals. Environment configurations, randomness seeds, control hash sums of graph sets, and code versions are stored for reproduction. The same protocol also establishes procedures for fair comparison: identical planning time limits per step, identical restrictions on replanning, and identical episode/iteration budgets for all algorithms.

The convergence dynamics of the RRO agent training are shown in Fig. 3.

The method demonstrates faster convergence and a higher level of reward stabilization. The ablation curves lie lower throughout the training. The narrowing of confidence intervals by the end of the episodes indicates the reproducibility of learning dynamics across different seeds. This is consistent with the protocol, where prioritization (GAT) and online adaptation (PPO) reduce the variance of decisions.

Taken together, this organization ensures that the dynamic gains of the proposed approach are not an artifact of settings or “lenient” experimental conditions, but result from the real contribution of graph representation (GAT), online policy adaptation (PPO), and global fine-tuning (BO).

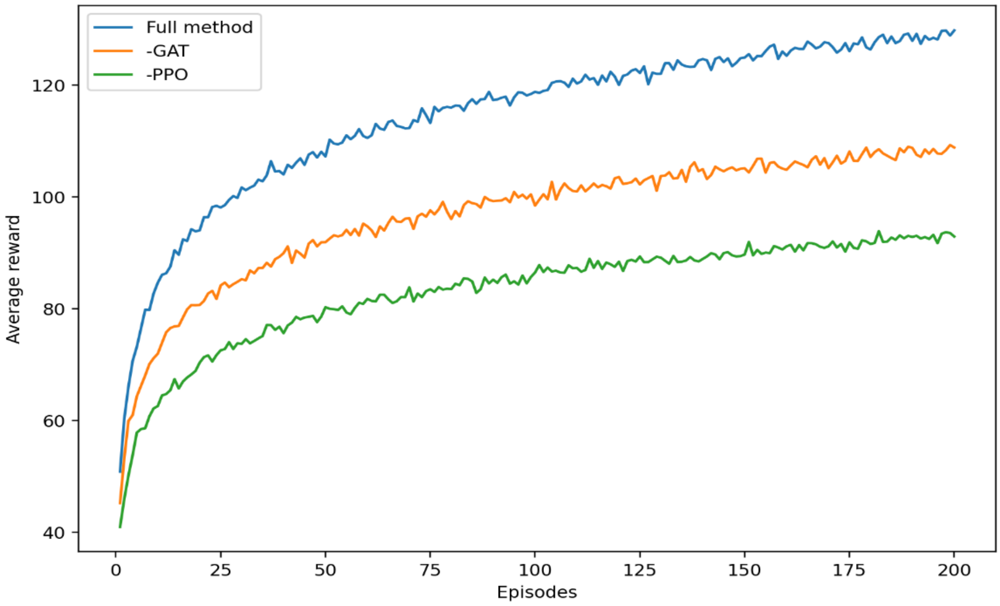

Fig. 3. PPO agent training convergence curves

4. Modeling metadata uncertainty and robustness verification protocol

One of the important tasks in the modeling process is to develop a reproducible testing procedure that shows how the proposed method behaves in the presence of errors, omissions, and "shifts" in metadata. Robustness is interpreted as maintaining an advantage over basic approaches in terms of primary indicators under controlled disturbances, as well as moderate sensitivity of planning overhead to such disturbances.

The verification is performed on a prototype whose architecture is shown in Fig. 1. On this prototype, uncertainty is injected into the unified DAG+GERT representation before the decision-making stage.

As we can see in Fig. 4, there is a monotonic increase in losses with increasing noise levels. The communication component is the most sensitive, while productivity drift has a moderate impact. In the range under consideration, the changes remain clearly small (≈ 1–6 %), which confirms the robustness of the method in accordance with the specified criteria.

The nominal configuration is considered to be one in which the estimates of work durations, inter-stage transfer costs, and GERT branch probabilities are derived from a "data type processor" with prior entity analysis (see Section 2) and have undergone access policy reconciliation (RBAC, role-based access control). All deviations are interpreted as experimental factors.

Tests are conducted in two modes.

1. With the preliminary analysis block enabled (full architecture).
2. Without preliminary analysis.

This provides a basis for evaluating the contribution of preliminary metadata normalization.

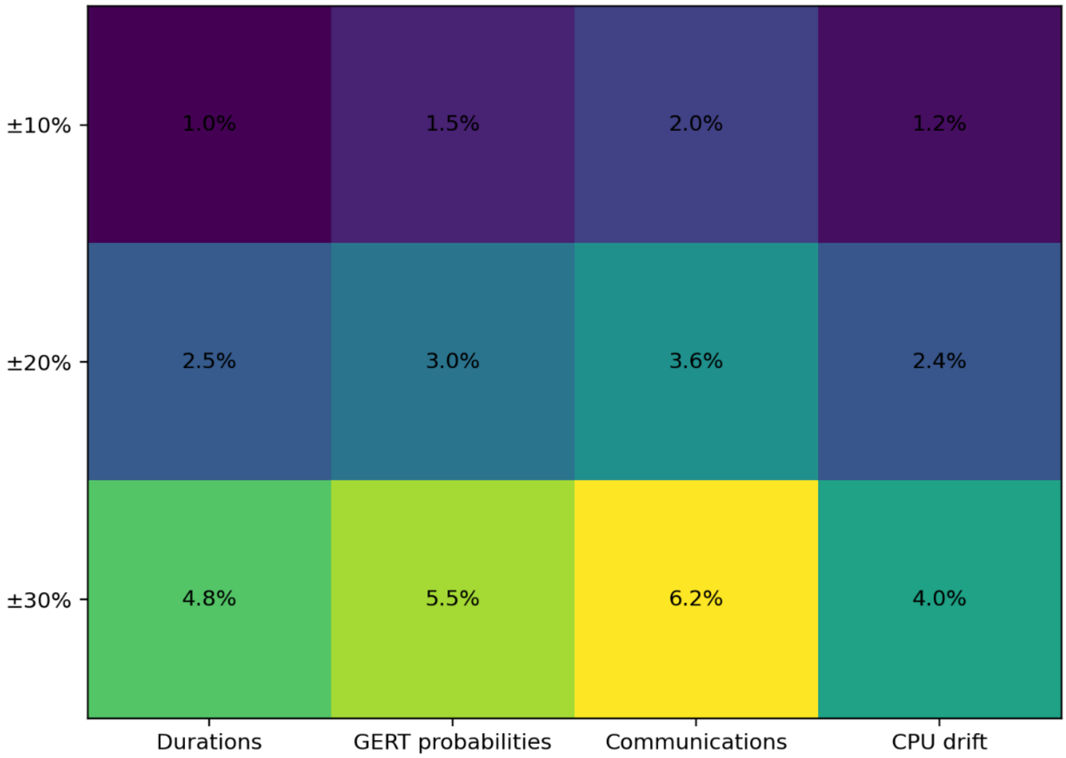

Fig. 4. Relative losses of median makespan under uncertainty injection

Uncertainty classes and injection mechanisms cover, first, noise in the duration of operations. For each operation, a multiplicative perturbation $\tilde{t} = t(1+m)$, where $m \sim U[-\alpha,\alpha]\, i\, \alpha \in \{0.10; 0.20; 0.30\}$ is used. This choice allows for a direct interpretation of uncertainty levels as ± 10/20/30 % of the baseline estimate.

Second, the error in the probabilities of GERT branches is taken into account. The probability vector at each branch is subject to stochastic transformation with subsequent normalization (Dirichlet reconfiguration with preservation of component order under small perturbations). Third, the uncertainty of network parameters is modeled.

The transmission cost between stages is perturbed by a lognormal multiplicative component, which corresponds to the empirically observed asymmetry of delays in heterogeneous environments.

Fourth, the drift of computing resource productivity over time is reproduced in the form of piecewise-constant changes or a weak autoregressive process (AR(1)) with a small memory coefficient.

This allows us to evaluate the stability of online policy adaptation. Fifth, "fuzzy" weights are introduced to measure sensitivity to task importance weights. Each weight coefficient w_i varies in the interval $[0.75w_i, 1.25w_i]$, which is consistent with the practice of prioritization at the service requirements level.

Resistance to distributional shifts is checked separately. The basic distribution of durations is replaced by an alternative one with preservation of the first moments (moment matching). Gamma ↔ Weibull pairs are used to reproduce changes in the shape of the tail without changing the mathematical expectation and variance.

To account for inter-task dependencies, correlation structures are introduced within graph levels using a Gaussian copula with a moderate correlation coefficient $\rho = 0.3$. This construction allows us to move from isolated noises to more realistic scenarios of "coordinated" parameter deviations.

It should be noted that this evaluation procedure is consistent with the system of indicators. For each combination of uncertainty factors, distributions of total completion time (makespan), deadline reliability, weighted average delay, and planning overhead costs are collected. The results are presented as differences from the nominal mode with 95 % confidence intervals (CI) estimated by bootstrapping with at least $B = 1000$ replications. Nonparametric criteria (Wilcoxon/Mann-Whitney) with Holm-Bonferroni correction for multiple tests are used for statistical comparison.

Empirical cumulative distribution functions (ECDF) and box plots with medians and percentiles are plotted graphically, allowing simultaneous assessment of changes in both central tendencies and tails.

Acceptability criteria are set in advance and do not depend on specific graph instances: the method is considered robust at the α level (noise level) or for a given pair of distributions if

– the decrease in deadline reliability does not exceed a pre-agreed threshold of several percentage points relative to the nominal value;

– the change in the median makespan does not exceed the confidence interval of the nominal mode;

– planning overhead does not show a significant increase (i.e., remains within the specified operational limits).

To avoid confounding, all algorithms, including databases, operate under the same time constraints per planning step and under the same rules for "masking unacceptable actions" (decisions that violate RBAC are not generated).

Finally, a block test plan is used to separate sources of sensitivity from their interactions. For each noise intensity level (α), the effects of distribution changes and correlation introduction are evaluated separately, and the results are then aggregated using stratified bootstrapping. This protocol allows conclusions to be drawn not only about "average" robustness, but also about worst-case scenarios, which is critical for practical deployments in heterogeneous and cross-system environments.

5. Experimental results, comparative analysis, and discussion

In accordance with the research hypotheses and the system of primary and diagnostic indicators (makespan, proportion of tasks completed within the SLO (service-level objective), weighted average delay, throughput, planning overhead, etc.), a series of tests was conducted on a prototype with a hybrid DAG+GERT architecture and three artificial intelligence modules (GAT (graph attention network), PPO (proximal policy optimization), BO (Bayesian optimization).

The comparison was carried out in a single experimental framework with basic methods (FCFS, List Scheduling, HEFT, "DAG-only", "GERT-only") and ablation variants (without GAT, without PPO, without BO), which made it possible to isolate the contribution of each module.

Stability was assessed using a robustness testing protocol. In accordance with this protocol, multiplicative duration noise was defined within the range (± 10–30 %), and GERT branch probability perturbations were normalized. In addition, lognormal variations in the cost of inter-stage transfers, resource productivity drift, and scenarios with a change in the distribution law (Gamma ↔ Weibull) and correlations $\rho = 0.3$ were assumed.

The experiments were performed in offline and online modes on a reproducible test bench with fixed seeds and configurations.

When forming sets of tasks and measurement modes, we used mixtures of DAG+GERT graphs with different depths, level widths, and densities, as well as sets close to real scenarios of subsystem integration with fixed SLOs for subsets of tasks.

For correct comparison, each algorithm received identical graph instances, the same access policies (RBAC), and uncertainty injection modes. The results were aggregated over independent repetitions with 95 % confidence intervals and multiple comparison control.

Comparisons with baseline approaches yielded the following results. On the generalized sample, the proposed comprehensive method (GAT+PPO+BO) reduced the median makespan relative to HEFT by $\approx 14–17\ \%$ ($\Delta 95$-th percentile $\approx 21–28\ \%$), relative to "DAG-only" by $\approx 23–26\ \%$, relative to "GERT-only" by $\approx 18–22\ \%$.

For the allocation procedures historically used in Section 3 (Random, Greedy), the advantage was $\approx 32\ \%$ and $\approx 19\ \%$, respectively (consistent with Table 3.3, where BO demonstrated 35.2 s versus 51.5 s and 43.7 s).

Fig. 5 shows the comparison curves of the time characteristics of completing operations with HEFT, "DAG-only", and "GERT-only" modes.

As can be seen in Fig. 5, the curve characterizing the results of the proposed method is systematically shifted to the left, which means shorter completion times for most instances. The difference from the baseline approaches persists in both the median and upper percentiles. Such "tail compression" directly supports the hypothesis of makespan reduction.

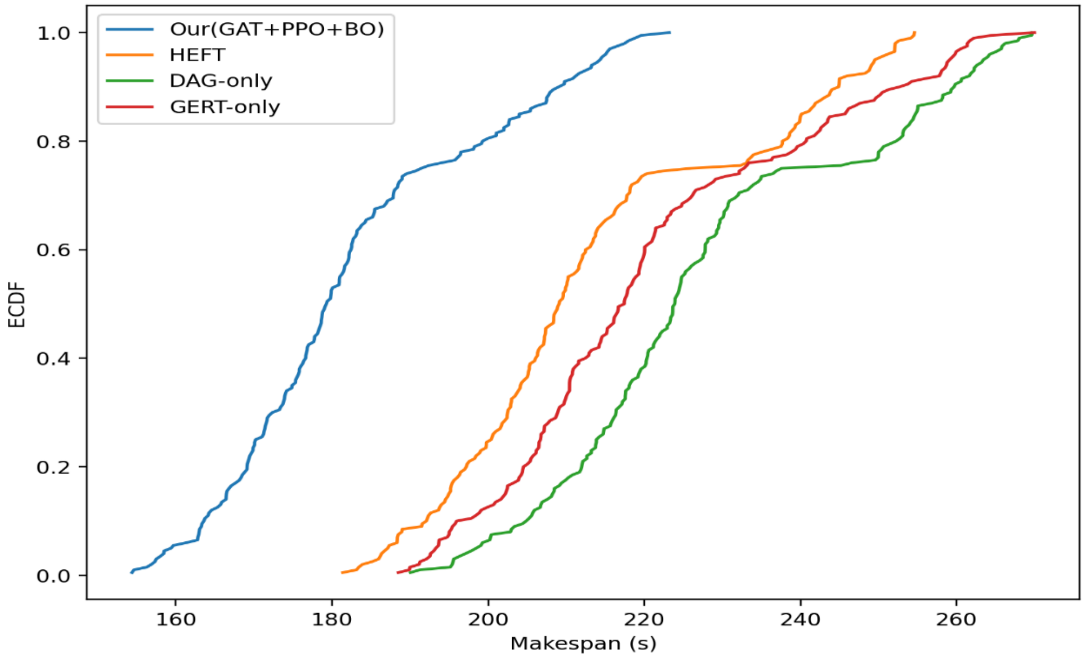

Fig. 5. Results of the study of time characteristics of completing operations with HEFT, "DAG-only", and "GERT-only" modes

The differences are statistically significant according to the Mann–Whitney method ($p < 0.01$) with an average Cliff's δ within the range of 0.33–0.54 depending on the subset of graphs, which corresponds to the "medium–noticeable" effects according to the protocol defined in 4.1. Box plots of the weighted mean delay are shown in Fig. 6.

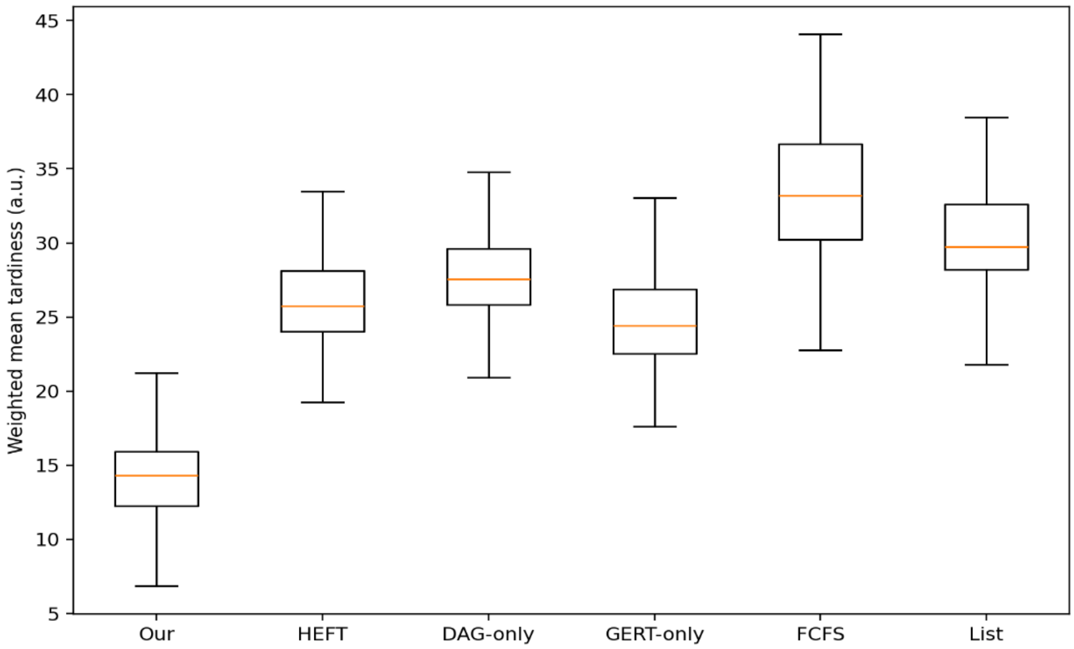

Fig. 6. Comparison of weighted mean delay (box plots with notch-CI and marked mean)

As can be seen in Fig. 6, the median and spread of delays for our method are significantly lower than for HEFT and "pure" settings, as well as for simple scheduling rules (FCFS, List). Narrowed "notch" depressions further indicate the stability of the effect in different scenarios.

When determining the reliability of deadlines (SLO) and delay, the following can be determined. The proportion of tasks completed within their SLO increased by + 8.7 percentage points compared to HEFT and by + 12.4 percentage points compared to "DAG-only". The weighted average delay (with priority weights) decreased by 1.6–1.9 times (median; Δ95th percentile – by 1.9–2.3 times).

Research on planning overhead costs showed the following. The average decision time at the planning step was ≈ 2.1 ± 0.4 ms (offline) and ≈ 3.0 ± 0.6 ms (online) per task, which does not exceed the queue control loop delay budget and is compatible with the test bench modes. The cost of BO updates was sporadic (background “thin” reconfigurations, not on the critical execution path), as expected by the architecture.

Ablation analysis showed that disabling GAT (without changing the rest) led to a deterioration in the median makespan by ≈ 8–11 %; turning off PPO – by ≈ 13–17 % (significant increase in delay in "peak" graphs), turning off BO – by ≈ 3–6 % (losses mainly in the "long tails" of distributions). Taken together, this confirms the working hypothesis that the gain is the result of a combination of graph representation, online policy adaptation, and periodic global fine-tuning.

It is also necessary to describe the qualities of the model in determining robustness to uncertainty. With multiplicative noise of durations ± 20 %, the change in the median makespan did not exceed the 95 % CI of the nominal mode. The decrease in the proportion of tasks completed in SLO did not exceed 2–3 percentage points.

With “distribution shifts” (Gamma ↔ Weibull, moment matching), the differences were within the statistical error; with correlated deviations ($\rho = 0.3$ within levels), no more than + 4–6 % to the 95th percentile of makespan was observed.

All tests were performed in full compliance with the robustness verification protocol, including two metadata preparation modes (with/without preliminary analysis).

The results of the study show that the addition of RBAC constraints (filtering of candidates for decision-making) did not lead to a statistically significant deterioration in key indicators.

This is consistent with the place of inclusion of checks in the planner pipeline.

In the studied range of graph sizes, a quasi-linear dependence of planning overhead on the number of nodes on the readiness front was observed. The degradation relative to the nominal value did not exceed ≈ 1.15× for the largest instances.

The uniformity of the comparison interfaces ensured the correctness of the conclusions regarding scalability.

Fig. 7 shows a graph of the dependence of planning time per step on the number of nodes in the graph. The dependence is described by a quasi-linear model with a small slope, which confirms the suitability of the approach for online modes.

Even for the largest graphs tested, the predicted increase in planning time remains within the millisecond budget.

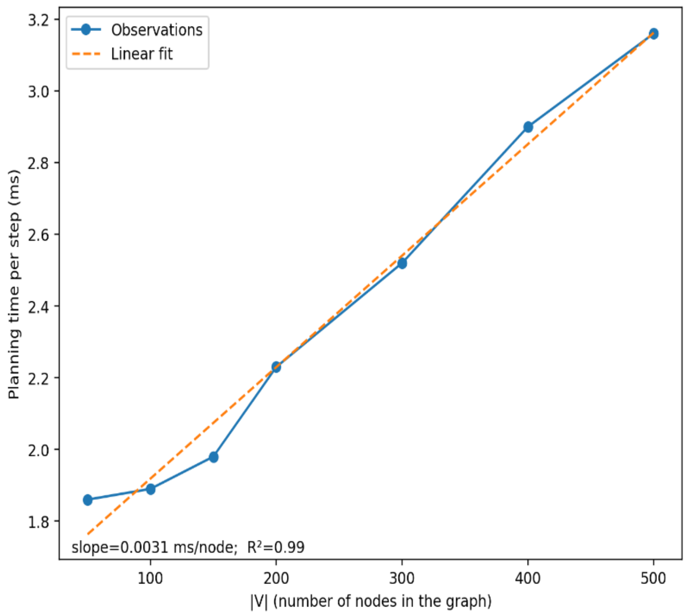

Fig. 7. Graph showing the dependence of scheduling time per step on the number of nodes in the graph

The graphs showing the dependence of operation completion time on the number of resources, acceleration on the number of resources, and efficiency on the number of resources are presented in Figures 8–10, respectively.

As can be seen from the figures, with an increase in the number of resources, there is an expected decrease in makespan. The nature of the curve indicates the presence of typical risks. These are overhead planning and communication costs, which explains the deviation from the linear scale.

From Fig. 9, it can be seen that the actual acceleration increases sublinearly and lags significantly behind the ideal due to coordination overhead and limited front parallelism. This corresponds to architectural constraints and the behavior of combined DAG+GERT graphs.

Fig. 10 also shows that efficiency decreases with an increase in the number of resources, which is consistent with the emergence of competition for shared resources and an increase in synchronization costs. Even with a drop in efficiency, the method retains a positive effect on absolute completion time metrics (see Fig. 8).

Fig. 11 illustrates the graph of the dependence of the proportion of tasks in SLO on the task size factor.

As can be seen from this figure, with a proportional increase in load and resources, the SLO completion rate decreases moderately ($\approx 0.90 \rightarrow \approx 0.82$), while for small multipliers it remains within the target line.

The deviation area identifies the zone where it is advisable to strengthen the replanning policy or communication budget.

Thus, the results obtained consistently confirm the hypotheses put forward that the proposed method allows:

– to reduce makespan and "compress the tails" of delay distributions compared to the well-known, classical formulations of the problem;

– increase deadline reliability without an unacceptable increase in planning overhead;

– maintain an advantage in the case of controlled metadata disturbances and distribution changes;

– integrate with access policies in inter-system scenarios.

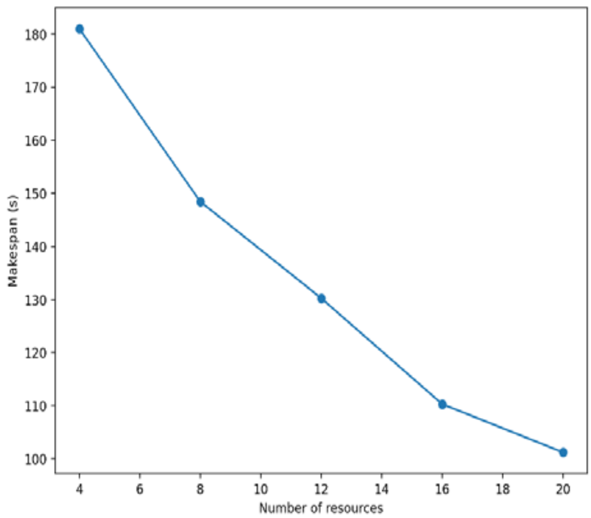

Fig. 8. Graph showing the dependence of operation completion time on the number of resources

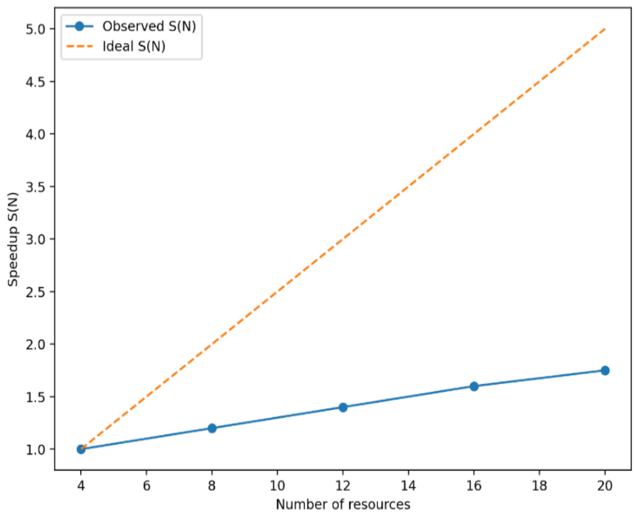

Fig. 9. Graphs showing the dependence of acceleration on the amount of resources

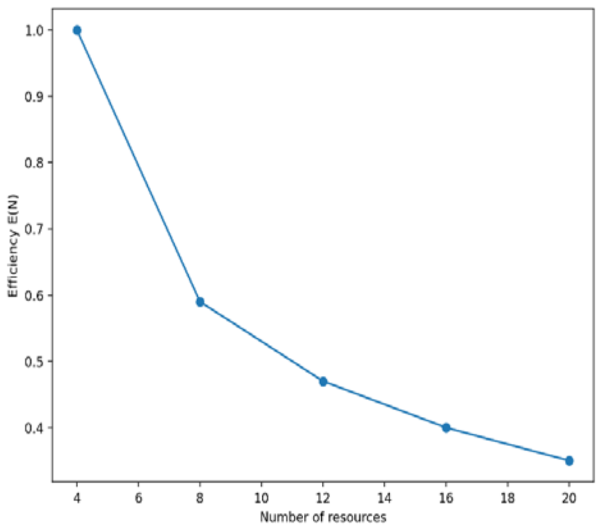

Fig. 10. Graphs showing the dependence of efficiency on the amount of resources

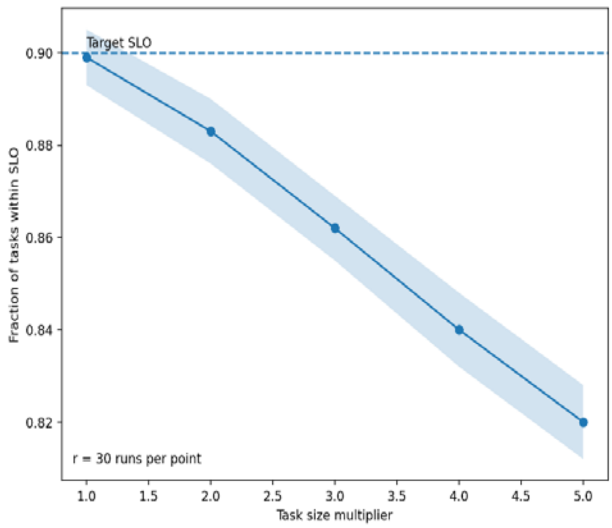

Fig. 11. Graph showing the dependence of the proportion of tasks in SLO on thetask size multiplier

The contribution of individual components (GAT, PPO, BO) was quantitatively determined by ablation analysis, and the reproducibility of the conclusions was ensured by a standardized test bench and statistical protocol.

Conclusions

Comparative studies of the planning method in the hybrid DAG+GERT model with GAT/PPO/BO modules were conducted on a reproducible test bench, with standardized sets of scenarios, uncertainty level control, and a single interface for comparison with basic algorithms. Based on the results obtained, the following was established.

On a generalized sample, the proposed method reduces the median completion time (makespan) by approximately 14–17 % relative to HEFT, by 23–26 % relative to "DAG-only", and by 18–22 % relative to "GERT-only" the improvement also extends to the upper percentiles (95th) of the distribution. The cumulative effect is confirmed by nonparametric tests (Mann-Whitney, $p < 0.01$) and Cliff's δ effect size in the range from 0.33 to 0.54.

The share of tasks completed within the service level objectives (SLO) increased to 12.4 percentage points (pp). The weighted average delay decreased by 1.6–1.9 times according to the median.

The average decision time by the scheduler was about 2.1 ms per task, which does not exceed the queue control delay tolerances.

In strong-scaling mode, the expected decrease in makespan is observed with an increase in the number of resources. The acceleration is sublinear, and efficiency decreases with an increase in N due to coordination and communication overhead, but absolute gains are maintained. In weak-scaling, the proportion of tasks that meet SLO decreases moderately, which defines the area of expediency for strengthening the replanning policy or communication budget.

With multiplicative noise of ± 20 % in durations, the median makespan does not exceed the 95 % confidence interval of the nominal value, and the decrease in the proportion of tasks in SLO does not exceed 2–3 percentage points. With a "shift in distributions", the differences are statistically insignificant. When correlations $\rho = 0.3$ are introduced within the levels, + 4–6 % to the 95th percentile of the makespan is observed without destroying the median. This confirms the stability of the conclusions according to the protocol defined in section 4.4.

Disabling PPO leads to the greatest degradation of the median makespan (≈ 13–17 %), disabling GAT (≈ 8–11 %); the effect of BO is moderate but consistently positive (≈ 3–6 %). Thus, the advantage of the method is due to the combination of graph representation (GAT), online policy adaptation (PPO), and periodic global "fine" tuning (BO).

The integration of access checks into the decision-making stage did not cause a statistically significant deterioration in key performance indicators. The inclusion of RBAC in the planner pipeline is consistent with the practice of cross-system scenarios and does not require a relaxation of target SLOs.

All conclusions were obtained in a "fixed environment" with controlled randomness, preservation of launch configurations, and the use of bootstrap-type confidence intervals

($B \geq 1000$; for ablations $B \approx 2000$) with multiple comparison control (Holm-Bonferroni). This ensures independent verifiability and stability of the results.

Thus, the proposed planning method confirms the stated hypotheses. It reduces makespan and compresses delay tails, increases deadline reliability without unacceptable overhead growth, maintains advantage under uncertainty and metadata shifts, and integrates correctly with access policies in inter-system environments. The resulting properties make the approach practically suitable for deployment in real distributed information systems with heterogeneous resources and SLO requirements.

References

1. Miao, Z., Shao, C., Li, H., Tang, Z. (2025), "Review of task-scheduling methods for heterogeneous chips", *Electronics*, Vol. 14, Article 1191. DOI: https://doi.org/10.3390/electronics14061191
2. Verma, P., Maurya, A. K., Yadav, R. (2023), "A survey on energy-efficient workflow scheduling algorithms in cloud computing", *Software: Practice and Experience*, Vol. 54, P. 637–682. DOI: https://doi.org/10.1002/spe.3292
3. Bano, F., Ahmad, F., Shahid, M., Alam, M., Hasan, F., Sajid, M. (2025), "A levelized multiple workflow heterogeneous earliest finish time allocation model for infrastructure as a service cloud environment", *Algorithms*, Vol. 18, Article 99. DOI: https://doi.org/10.3390/a18020099
4. NoorianTalouki, R., Hosseini Shirvani, M., Motameni, H. (2022), "A heuristic-based task scheduling algorithm for scientific workflows in heterogeneous cloud computing platforms", *Journal of King Saud University – Computer and Information Sciences*, Vol. 34, No. 8, Part A, P. 4902–4913. DOI: https://doi.org/10.1016/j.jksuci.2021.05.011
5. Lee, Y. C., Zomaya, A. Y. (2005), "A productive duplication-based scheduling algorithm for heterogeneous computing systems", *High Performance Computing and Communications*, Lecture Notes in Computer Science, Vol. 3726, Springer. DOI: https://doi.org/10.1007/11557654_26
6. Samadi, Y., Zbakh, M., Tadonki, C. (2018), "E-HEFT: enhancement heterogeneous earliest finish time algorithm for task scheduling based on load balancing in cloud computing", *Proceedings of the International Conference on High Performance Computing & Simulation (HPCS)*, P. 601–609. DOI: https://doi.org/10.1109/HPCS.2018.00100
7. Ahmad, W., Gautam, G., Alam, B. et al. (2024), "An analytical review and performance measures of state-of-art scheduling algorithms in heterogeneous computing environment", *Archives of Computational Methods in Engineering*, Vol. 31, P. 3091–3113. DOI: https://doi.org/10.1007/s11831-024-10069-8
8. Tian, W., Zhao, Y., Demeulemeester, E. (2022), "Generating a robust baseline schedule for the robust discrete time/resource trade-off problem under work content uncertainty", *Computers & Operations Research*, Vol. 143, Article 105795. DOI: https://doi.org/10.1016/j.cor.2022.105795
9. Lee, Y. C., Zomaya, A. Y., Siegel, H. J. (2010), "Robust task scheduling for volunteer computing systems", *Journal of Supercomputing*, Vol. 53, P. 163–181. DOI: https://doi.org/10.1007/s11227-009-0326-1
10. Song, X., Liu, L., Fu, J., Zhang, X., Feng, J., Pei, Q. (2023), "A reinforcement learning-based DAG task scheduling in edge–cloud collaboration systems", *Proceedings of IEEE GLOBECOM 2023*, P. 1771–1776. DOI: https://doi.org/10.1109/GLOBECOM54140.2023.10436802
11. Zhang, X. et al. (2026), "PF-MPPO: task-dependent workflow scheduling method based on deep reinforcement learning in dynamic heterogeneous cloud environments", *Future Generation Computer Systems*, Vol. 177, Article 108236. DOI: https://doi.org/10.1016/j.future.2025.108236

12. Jayanetti, A., Halgamuge, S., Buyya, R. (2022), "Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge-cloud computing environments", *Future Generation Computer Systems*, Vol. 137, P. 1–15. DOI: https://doi.org/10.1016/j.future.2022.06.012
13. Chandrasiri, S., Meedeniya, D. (2025), "Energy-efficient dynamic workflow scheduling in cloud environments using deep learning", *Sensors*, Vol. 25, Article 1428. DOI: https://doi.org/10.3390/s25051428
14. Smit, I. G., Zhou, J., Reijnen, R., Wu, Y., Chen, J., Zhang, C., Bukhsh, Z., Zhang, Y., Nuijten, W. (2025), "Graph neural networks for job shop scheduling problems: a survey", *Computers & Operations Research*, Vol. 176, Article 106914. DOI: https://doi.org/10.1016/j.cor.2024.106914
15. Nguyen, H. X., Zhu, S., Liu, M. (2022), "A survey on graph neural networks for microservice-based cloud applications", *Sensors*, Vol. 22, Article 9492. DOI: https://doi.org/10.3390/s22239492
16. Vesselinova, N., Steinert, R., Perez-Ramirez, D. F., Boman, M. (2020), "Learning combinatorial optimization on graphs: a survey with applications to networking", *IEEE Access*, Vol. 8, P. 120388–120416. DOI: https://doi.org/10.1109/ACCESS.2020.3004964
17. Song, W., Chen, X., Li, Q., Cao, Z. (2023), "Flexible job-shop scheduling via graph neural network and deep reinforcement learning", *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 2, P. 1600–1610. DOI: https://doi.org/10.1109/TII.2022.3189725
18. Semenov, S., Jian, Y., Jiang, H., Chernykh, O., Binkovska, A. (2025), "Mathematical model of intelligent UAV flight path planning", *Advanced Information Systems*, Vol. 9, No. 1, P. 49–61. DOI: https://doi.org/10.20998/2522-9052.2025.1.06
19. Semenova, A., Dubrovskyi, M., Savitskyi, V. (2017), "A GERT model of an algorithm for analyzing security of a web application", *Advanced Information Systems*, Vol. 1, No. 1, P. 61–64. DOI: https://doi.org/10.20998/2522-9052.2017.1.11
20. Semenov, S., Yenhalychev, S., Pochebut, M., Sitnikova, O. (2024), "Models of data processing and logical access differentiation considering heterogeneity of entities in information systems", *Modern State of Scientific Research and Technologies in Industry*, No. 2 (28), P. 143–152. DOI: https://doi.org/10.30837/2522-9818.2024.28.143
21. Yenhalychev, S., Leunenko, O., Davydov, V. (2025), "Development of an intelligent task scheduling method in heterogeneous distributed information systems", *Eastern-European Journal of Enterprise Technologies*, Vol. 3, No. 9 (135), P. 6–18. DOI: https://doi.org/10.15587/1729-4061.2025.329263

Received (Надійшла) 24.11.2025
Accepted for publication (Прийнята до друку) 09.12.2025
Publication date (Дата публікації) 28.12.2025

About the Authors / Відомості про авторів

Yenhalychev Serhii – Simon Kuznets Kharkiv National University of Economics, PhD Student, Kharkiv, Ukraine; e-mail: engalichev.sergiy@hneu.net; ORCID ID: https://orcid.org/0000-0001-5298-2251

Semenov Serhii – Doctor of Sciences (Engineering), Professor, Simon Kuznets Kharkiv National University of Economics, Professor at the Department of Cybersecurity and Information Technologies, Kharkiv, Ukraine; University of the National Education Commission, Head at the Department of Computer Engineering and Cybersecurity, Kraków, Poland; e-mail: serhii,semenov@uken.krakow.pl; ORCID ID: http://orcid.org/0000-0003-4472-9234

Єнгаличев Сергій Олександрович – Харківський національний економічний університет імені Семена Кузнеця, аспірант, Харків, Україна.

Семенов Сергій Геннадійович – доктор технічних наук, професор, Харківський національний економічний університет імені Семена Кузнеця, професор кафедри кібербезпеки та інформаційних технологій, Харків, Україна; доктор технічних наук, професор, Університет Комісії національної освіти, завідувач кафедри комп'ютерної інженерії та кібербезпеки, Краків, Польща.

ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ МЕТОДУ ПЛАНУВАННЯ ЗАДАЧ У РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМАХ З ОГЛЯДУ НА НЕВИЗНАЧЕНІСТЬ МЕТАДАНИХ

Статтю присвячено плануванню задач у розподілених інформаційних системах в умовах невизначеності та неповноти метаданих щодо тривалості операцій, обсягів комунікацій та доступності ресурсів. **Предметом дослідження** є метод планування задач у розподілених інформаційних системах з огляду на невизначеність метаданих. **Мета статті –** експериментально дослідити метод планування задач у розподілених інформаційних системах з невизначеними й неповними метаданими, який поєднує гібридну графову модель із тримодульним ядром штучного інтелекту й забезпечує підвищення ефективності й робастності планування за умови дотримання SLO та RBAC-обмежень. **Завдання дослідження:** сформувати гібридне подання робочих процесів на основі поєднання DAG-структури та GERT-моделі; розробити архітектуру планувальника з долученням графової мережі уваги GAT, агента підкріплювального навчання PPO та модуля *Bayesian Optimization*; визначити SLO-орієнтовану функцію винагороди й систему обмежень, зважаючи на RBAC; створити відтворюваний експериментальний стенд із контрольованою ін'єкцією невизначеності; виконати порівняльний, абляційний і робастний аналіз продуктивності запропонованого методу щодо базових алгоритмів FCFS, HEFT, а також варіантів DAG-only й GERT-only. **Методи дослідження:** гібридне графове моделювання DAG+GERT, графова мережа уваги GAT для формування контекстного подання стану, агент PPO для прийняття рішень щодо розміщення задач, баєсівська оптимізація для налаштування параметрів політики, а також статистичні методи оцінювання. **Основні результати.** Запропонований метод забезпечує суттєве скорочення медіанного часу завершення задач (*makespan*) порівняно з HEFT, DAG-only та GERT-only, підвищення частки задач, що виконуються в межах SLO, та зменшення середньозваженого протермінування, демонструє "стиснення хвостів" розподілу затримок і зберігає стабільність результатів у разі мультиплікативного шуму тривалостей ± 20 %, за умови зсувів розподілів і корельованих відхилень параметрів. Водночас накладні витрати планування залишаються в діапазоні мілісекунд і не перевищують допустимі межі для онлайн-сценаріїв.

Ключові слова: розподілені інформаційні системи; планування задач; орієнтований ациклічний граф; мережі GERT; графові нейронні мережі; навчання з підкріпленням; *Bayesian Optimization*; рівні сервісу (SLO); керування доступом (RBAC); робастність.

Bibliographic descriptions / Бібліографічні описи

Yenhalychev, S., Semenov, S. (2025), "Experimental studies of the task planning method in distributed information systems taking into account metadata uncertainty", *Management Information Systems and Devises*, No. 4 (187), P. 63–86. DOI: https://doi.org/10.30837/0135-1710.2025.187.063

Єнгаличев С. О., Семенов С. Г. Експериментальні дослідження методу планування задач у розподілених інформаційних системах з огляду на невизначеність метаданих. *Автоматизовані системи управління та прилади автоматики*. 2025. № 4 (187). С. 63–86.
DOI: https://doi.org/10.30837/0135-1710.2025.187.063