Розглядається проблема побудови web-орієнтованої системи підтримки прийняття рішень при плануванні виконання договорів на підприємствах, які діють у сфері надання послуг. Досліджуються характерні особливості діяльності таких підприємств. Побудовано удосконалену математичну модель задачі планування виконання договорів та визначено критерії оцінки ефективності отриманого варіанту рішення. У якості оцінки ефективності варіанту виконання запропоновано використання адитивної згортки часткових критеріїв з встановленням пріоритету виконання кожного з них. Розроблено комбінований алгоритм для вирішення задачі планування, направлений на врахування особливостей предметної області. Для використання комбінованого алгоритму вхідні дані запропоновано представити у вигляді багатошарового графа, кожний шар якого відповідає робочому часу одного виконавця, а вузли визначають моменти синхронізації між шарами. При розробці комбінованого алгоритму використано елементи АСО алгоритму та генетичного алгоритму. Запропоновано структуру СППР при плануванні виконання договорів, а також розглянуто технології, використані для її практичної реалізації. Наведено приклади інтерфейсу користувача системи. СППР забезпечує підтримку функцій управління при плануванні та дозволяє здійснювати глибокий аналіз ситуацій, їх оцінку та вибір оптимального варіанту календарного плану, виконуючи усі підготовчі дії та формуючи вже готові варіанти рішень. Створена система є web-орієнтованою, що дозволяє використовувати її в будь-якому місці, де є доступ до мережі Інтернет

Ключові слова: web-орієнтована система підтримки прийняття рішень, планування виконання договорів, комбінований алгоритм, програмний модуль

UDC 004.9:004.02:65.012.2
DOI: 10.15587/1729-4061.2018.132604

WEB-ORIENTED DECISION SUPPORT SYSTEM FOR PLANNING AGREEMENTS EXECUTION

S. Hrybkov
PhD, Associate Professor*
E-mail: sergio_nuft@nuft.edu.ua
H. Oliinyk
Postgraduate student*
E-mail: hanna_oliinyk@nuft.edu.ua
V. Litvinov
Doctor of Technical Sciences, Professor, Lead Researcher
Institute of Mathematical Machines and Systems Problems of NASU
Akademika Hlushkova ave., 42, Kyiv, Ukraine, 03187
E-mail: L39143O6@gmail.com
*Department of Information Systems
National University of Food Technologies
Volodymyrska str., 68, Kyiv, Ukraine, 01601

1. Introduction

Activities of most modern enterprises are associated with rendering different services in accordance with concluded agreements. Such activity is characterized by certain specific aspects that justify its distinguishing among others, such as, for example, manufacturing certain products. First, the sequence of implementation of stages of each agreement, their number and time required to perform them, depend both on the features of the general type of a service and the requirements of a particular customer, which are determined by the terms of an agreement. Second, the implementation of stages of one agreement can be carried out by different executors simultaneously provided this does not violate logical consistency of works. Third, the need to provide resources for the stages, such as certain technical equipment, depends on the customer needs. Fourth, the total number of agreements to be implemented over a planned period is not regulated and is impossible to determine in advance because orders can be placed at different times for both different and the same kinds of services, but with different individual requirements. The placement of new orders requires operative changes to the existing calendar plan, as well as setting such execution terms that will produce the least impact on other works. The task of agreements execution planning for most enterprises relates to the class of multi-criteria combinatorial problems. It is important to note that, in order to find a solution, it is not always necessary to gain maximal profits only, but to coordinate many factors and take into consideration constraints as well. In addition, in the process of solving a problem on planning, it is impossible to avoid taking additional coordinating decisions, especially in the case of certain deviations from established norms [1].

Planning the execution of work relates to various areas of activity, including management of projects with different complexity: from rendering small services and organizing holidays to designing and constructing spacecraft [2]. Even though the methods for solving similar problems were considered by various scholars using classical, heuristic, and evolutionary methods, it has remained relevant because each subject area has its own characteristics [3, 4]. The relevance of the search for new approaches and procedures to solve a given problem is predetermined by the development of heuristic and evolutionary methods, which in turn relates to the rapid development of information technologies in general [5].

2. Literature review and problem statement

In paper [1], authors proposed a mathematical model of the problem on constructing a plan for agreements execution for businesses, the activity of which relates to rendering services and conducted a review of existing approaches and methods for solving such problems. As a result of the studies conducted, it was found that the developed mathematical model, shown in [1], is incomplete and needs some improvement, in particular, the introduction of an additional criterion for effectiveness evaluation. According to the results of mathematical modeling, the general criterion for evaluation of the effectiveness of a variant of implementation of the phases of agreements was separated and a conclusion on the feasibility of using ACO (Ant Colony Optimization) algorithm for solving the problem on agreements execution planning was made.

Authors of paper [2] considered an approach to solving the problem on planning with limited resources based on the Particle Swarm Optimization method to minimize costs. In the paper, according to results of the conducted studies, the conclusion about the appropriateness of using this approach for a wide range of problems of planning, in particular in project management, was made. However, its application is possible only if a problem is reduced to more general terms; in addition, the method is not aimed at minimizing the total time of works execution and does not include a time limit.

Authors of research [3] compared several heuristic algorithms but did not give examples of their application for solving specific management problems.

Authors of work [4] consider possibilities of the practical application of genetic algorithms for solving problems on the optimization of control and planning industrial processes. But the work has a generalized character and does not take into account the features of each specific area.

Paper [5] offers a hybrid algorithm for planning based on the Particle Swarm Optimization and the ACO algorithm, which allows determining the best sequences for different combinations of delays in execution and warnings for a given a set of works. Fuzzy logic is used to determine the optimal variant, which largely satisfies the given objective function. The authors of work [5] also offer a comparison of the algorithm, proposed by them, with different modifications of the genetic algorithm, specifically: a partial representation of the crossover operator, the use of multi-component unified serial generator, the use of fuzzy genetic algorithm. The work is aimed more at minimizing delays between implementation of stages, besides the results of the proposed method were tested on the classic statement of the problem of schedule making with the use of parallel machines without taking into account economic and social influences in finding the optimal plan.

Using the basic ideas of ACO algorithm, a combined algorithm was created, the application of which provides the possibility of finding variants of solutions, close to the optimum over minimum time [6]. Experimentally it was found that at an increase in the dimensionality of a problem, the combined algorithm gives a better result in comparison with the genetic algorithm and the classic ACO. Testing the developed algorithm on real data samples made it possible to reveal shortcomings, which also applies to programming implementation.

Paper [7] presented a mathematical model for the plan of operation execution at different technological machines that can perform operations in parallel, with different periods of planning and a varying number of parallel machines or executors. But only the following criteria were considered as evaluation criteria: minimization of time of the last work completion, minimization of the total delay, minimization of the total time of execution in advance.

To solve the problem of planning, authors of publication [8] presented the approach based on the Artificial Bee Colony algorithm using the local search method to enhance the operational capability of the basic algorithm. Besides, an additional neighborhood operator based on a greedy constructive-destructive procedure was considered. In addition, the proposed algorithm was compared with such algorithms as the search with restrictions and the genetic algorithm. However, the use of the proposed algorithm is possible only for problems that have minimum variations of the course of events without taking into account priorities of partial criteria and restrictions.

In paper [9], improvement of effectiveness of the genetic algorithm by using the selective initialization mechanism was proposed. The problem of dependence of the speed of finding a result and its proximity to the optimal one on selection of the primary population of the algorithm was considered. The authors proposed a new approach to enhancement of the effectiveness of the genetic algorithm of selective initialization, which is aimed at finding the most appropriate "chromosomes" and using them as a source for the genetic algorithm. Using a modified genetic algorithm is only possible in complex to improve another method, which will be the basic one. Moreover, application of the genetic algorithm is possible when using the information system that will ensure implementation of all functions.

Authors of paper [10] proposed a new hybrid algorithm based on a multi-purpose genetic algorithm with the use of simulation annealing for parallel machines with dependent execution sequences, different deadlines, and established priorities. Even though the proposed algorithm cannot be applied without the use of information technologies, its drawback is complexity of adaptation of different input data.

Authors of work [11] explore optimization problems on the graphs of large dimensionality, for which there is no effective algorithms of a search for an accurate solution. Specifically, such problems as formation of the optimal schedule for the agents who serve clients, the problem of designing the optimal route for a car park are considered. The mathematical models, methods of decomposition into simpler sub-problems, an approach to organization of data storage in memory, algorithms for solving, assessment of operation time and the used memory amount are presented for such tasks.

The conducted analysis of the above literary sources [1–11] allows us to argue about appropriateness of improvements and the use of the combined algorithm as a mathematical basis for decision support systems (DSS) when planning agreements execution. It is advisable to use the ACO algorithm as a basic algorithm. It is necessary to improve it with the use of genetic algorithms, which will ensure finding optimal solutions when working with graphs of large dimensionality, describing variability of orders execution. In addition, based on the scientific literature [6, 9, 10], there is a possibility to modify each of the algorithms, which will allow the improvement of the general algorithm at the expense of separate components. Given the complexity of the problem regarding agreements execution planning, it is necessary to provide the possibility of clarification and correction of input and intermediate data to obtain different variants of the plan with their subsequent assessment. It is DSS that will allow us, using the automation of intelligent work, to obtain the possibility to design, compare and choose alternative options of solutions for employees of the planning department and production units with the use of computing tools. In addition, only a system that is fully adapted to the requirements of a given subject area will ensure optimum solution to the problem on planning.

3. The aim and objectives of the study

The aim of present research is to develop a web-oriented DSS for planning agreements execution at enterprises that operate in the service rendering business to ensure the efficiency/quality of work of employees at a planning department and at production units.

To accomplish the aim, the following tasks have been set:

– to develop a mathematical model for agreements execution planning regarding the general economic and social influences while finding the optimal plan;

– to create an algorithm for solving a problem on agreements execution planning, which would implement the appropriate model and ensures minimization of time cost for finding the optimal solution;

– to design a DSS structure and to choose the technologies and means for development of its components, which would ensure solving a problem on agreements execution planning in online mode.

4. Development of the mathematical model for agreements execution planning

The main problem of making decisions when planning agreements execution is to construct and select an optimal variant of execution of orders. In this case, all orders that are already being fulfilled at an enterprise, as well as those that have just arrived, are taken into account. The optimal variant of execution is considered to be the one that ensures minimization of consumption of resources and time to fulfill all orders, as well as the maximal load of all executors. A special feature of the problem of calendar planning is that the existing calendar plan is modified in the process of its implementation. This problem refers to the class of multicriteria optimization problem because it requires the use of different assessment selection criteria, each of which directly affects the choice of the optimal solution. It should be noted that in any situation, there are several permissible variants of execution, from which it is necessary to choose the best one. The additive convolution is used to derive the generalized estimate of the implementation plan. In this case, there is a need to take into account not only every separate partial criterion, but also to set priorities for each of them in accordance with a particular case. Additive convolution is represented by the formula (1) and is aimed at the minimization of value [1].

$$F_0 = \sum_{\phi=1}^{\Phi} \xi_\phi * \lambda_\phi \to \min, \tag{1}$$

where ξ_ϕ is the indicator of priority factor of each partial criterion of assessment of the effectiveness of the variant of execution.

$$0 \le \xi_\phi \le 1, \quad \sum_{\phi=1}^{\Phi} \xi_\phi = 1;$$

ϕ is the number of a criterion ($1 \le \phi \le \Phi$); Φ is the total number of partial criteria.

λ_ϕ is the normalized φ-th partial criterion; normalization is performed with the aim of reducing different criteria to a unified dimensionality and is calculated from formula (2):

$$\lambda_\phi = \frac{F_\phi - F_\phi^{\min}}{F_\phi^{\max} - F_\phi^{\min}}, \tag{2}$$

where F_ϕ is the value of the ϕ-th partial criterion; $F_\phi^{\min}$ is the minimal value of the ϕ-th partial criterion; $F_\phi^{\max}$ is the maximal value of the ϕ-th partial criterion.

It is possible to evaluate the quality and effectiveness of the resulting variant of the calendar plan using different criteria, the use of which is determined by specific conditions. The authors distinguish five main partial criteria for the problem of agreement execution planning:

– minimization of total time for execution;

– minimization of total time of execution delay;

– minimization of total amount of fines for execution delay;

– minimization of total costs for execution of agreements within the given period;

– minimization of downtime of executors.

The first criterion is minimization of total time for execution of all agreements of the planned period (3):

$$F_1 = \sum_{i=1}^{n} (u_i * Z_i) \to \min, \tag{3}$$

where i is the number of the agreement to be executed within the planned period; n is the total number of agreements to be executed within the planned period; u_i is the coefficient of significance of execution of the i-th agreement, for which the following conditions exist:

$$0 \le u_i \le 1, \quad \sum_{i=1}^{n} u_i = 1;$$

Z_i is the time of execution of the i-th agreement, on condition of taking into consideration of the sequence of stages with their possible implementation.

The time of agreement execution is equal to the difference between the maximal moment of completion and minimal moment of beginning of implementation, which is described by the formula (4):

$$Z_i = \max_j \left(\sum_{l=1}^{m} \left(\left(tp_{ijl} + td_{ijl} + tv_{ijl} \right) * y_{ijl} \right) \right) - \min_j \left(\sum_{l=1}^{m} \left(tp_{ijl} * y_{ijl} \right) \right), \tag{4}$$

where i is the number of an agreement to be executed within the planned period; j is the number of a stage from the set of stages, $j \in J_i$ is the set of stages for the i-th agreement $J_i = \{1, 2, \ldots, \omega_i\}$, ω_i is the number of stages of the i-th agreement; l is the number of an executor from the set of executors, $l \in M$; $M = \{1, 2, \ldots, m\}$ is the set of executors, involved in execution of works; m is the number of executors, involved during the planned period; tp_{ijl} is the time of the beginning of implementation of the j-th stage of the i-th agreement by the l-th executor; td_{ijl} is the time of waiting before implementation of the j-th stage of the i-th agreement by the l-th executor in regard to implementation of the previous stages of the i-th agreement; tv_{ijl} is the time of implementation of the j-th stage of the i-th agreement of the l-th executor; y_{ijl} is the parameter that takes the values {0, 1} (y_{ijl}=1, if the j-th stage of the i-th agreement is implemented by the l-th executor; y_{ijl}=0 in another case).

Another partial criterion of assessment of effectiveness of the variant of the agreement execution plan is the total time of delay in execution of all agreements within the planned period, calculated from (5):

$$F_2=\sum_{i=1}^{n}\left(u_i*\Psi_i\right)\to\min, \tag{5}$$

where i is the number of an agreement to be executed within the planned period; n is the number of agreements to be executed within the planned period; u_i is the coefficient of significance of execution of the i-th agreement, for which the following conditions exist:

$$0\le u_i\le 1,\ \ \sum_{i=1}^{n}u_i=1;$$

Ψ_i is the total time of execution of the i-th agreement that is determined from formula (6):

$$\Psi_i=\begin{cases}Z_i-T_i, & \text{if}\ \ Z_i>T_i,\\ 0, & \text{if}\ \ Z_i\le T_i,\end{cases} \tag{6}$$

where T_i is the directive term of execution of the i-th agreement, determined by the conditions of agreement $Tp\le T_i\le Tz$ (Tp is the time of the beginning of the planned period, Tz is the time of completion of the planned period; Z_i is the time of execution of the i-th agreement.

The third criterion is the total amount of the fines for the delay of agreement execution for the planned period. This criterion is determined from the formula (7):

$$F_3=\sum_{i=1}^{n}\left(u_i*g_i*\Psi_i\right)\to\min, \tag{7}$$

where i is the number of an agreement to be executed within the planned period; n is the number of agreements to be performed within the planned period; u_i is the coefficient of significance of execution of the i-th agreement, for which the following conditions exist:

$$0\le u_i\le 1,\ \ \sum_{i=1}^{n}u_i=1;$$

g_i is the amount of penalties at violation of directive term T_i of execution of the i-th agreement; Ψ_i is the total time of execution of the i-th agreement, which is determined from formula (6).

The fourth criterion is the value of total costs for execution of agreements within the assigned period, which is calculated from the formula (8):

$$F_4=\sum_{i=1}^{n}\left(u_i*\left(\sum_{j=1}^{\omega_i}\sum_{l=1}^{m}\left(\left(td_{ijl}+tv_{ijl}\right)*y_{ijl}*c_{ijl}\right)\right)\right)\to\min, \tag{8}$$

where i is the number of an agreement to be executed within the planned period; n is the number of agreements to be executed within the planned period; u_i is the coefficient of significance of execution of the i-th agreement, for which the following conditions exist:

$$0\le u_i\le 1,\ \ \sum_{i=1}^{n}u_i=1;$$

j is the number of a stage from the set of stages, ω_i is the number of stages of the i-th agreement; l is the number of an executor form the set of executors, $l\in M$; m is the number of executors, engaged during the planned period; td_{ijl} is the time of waiting before implementing of the j-th stage of the i-th agreement by the l-th executor in regard to implementation of the previous stages of the i-th agreement; c_{ijl} is the cost of implementation of the j-th stage of the i-th agreement by the l-th executor within an hour; y_{ijl} is the parameter that takes the values {0,1} ($y_{ijl}=1$, if the j-th stage of the i-th agreement is executed by the l-th executor; $y_{ijl}=0$ in another case).

The fifth criterion is minimization of the total downtime or the time of inactivity of executors (9):

$$F_5=\sum_{j_l=1}^{\omega_l}\sum_{l=1}^{m}\left(tzl_{(j-1)_l l}-tpl_{(j)_l l}\right)\to\min, \tag{9}$$

where i is the number of agreement to be executed within the planned period; n is the total number of agreements to be executed within the planned period; j_l is the number by order of the stage that is implemented by l-th executor, $j_l\in J_l$, J_l={1, 2,..., ω_l}, ω_l is the number of works, executed by the l-th executor; l is the number of the executor from the set of executors; m is the number of executors, engaged during the planned period; $tzl_{(j-1)_l l}$ is the time of implementation of the $(j-1)_l$-the stage, l is the number of an executor from the set of executors, $l\in M$; $tpl_{(j)_l l}$ is the time of the beginning of implementation of the j_l-the stage; l is the number of an executor from the set of executors, $l\in M$.

Taking into account the features of the problem, a whole range of constraints and additional conditions are used [1].

It is assumed that each l-*th* executor can simultaneously implement not more than one j-stage and of the i-th agreement

$$\sum_{l\in M}y_{ijl}=1.$$

All stages of all agreements must be implemented, which is a prerequisite for rendering services to customers in a full range

$$\sum_{j\in J_i}\sum_{l\in M}y_{ijl}=\omega_i.$$

The beginning of execution of any work by the l-th executor is possible only after completion of execution by the same executor of the previous work $tp_{ijl}\ge tp_{qvl}$, $q\in N$, $v\in J_q$, $i\ne q$. On condition that the j-th stage of the i-th agreement is impossible to implement without completion of implementation of the previous ones

$$tp_{ijl}\ge\max_h\left(tp_{ihs}+tv_{ihs}+td_{ihs}\right),\ \ s\in M,\ \ h\in J_i,\ \ h<j.$$

The time of the beginning of implementation of the j-th stage of the i-th agreement by the l-th executor will come only when the previous h-th stages are implemented by the s-th executor. In this case, to determine the rest of the time for completing the implementation of such stages, it is sufficient to select the one that is the last to finish. If the next stage can be implemented regardless the execution of the previous ones, restrictions are not met: $tp_{ijl} \geq 0$.

The need for maximum continuity of execution and a decrease in downtime that relate to the lack of resources cause timely provision of each stage with necessary resources.

The totality of resources that are necessary for execution of the i-th agreement is calculated from (10):

$$R_i = \sum_{j=1}^{\omega_i} r_{ij}, \tag{10}$$

where r_{ij} is the totality of resources, required for implementation of the j-th stage of the i-th agreement.

Provision of resources directly influences timely and faultless execution of the agreement $tr_{ij} \leq tp_{ijl}$.

The constructed mathematical model shows that the problem of agreement execution planning belongs to the general class of NP-complex multi-criteria combinatorial problems. Dimensionality of this problem is determined by the number of elements of sets N, J, M.

To solve the problem on agreements execution planning, it is proposed to use the combined algorithm [6], which uses the ideas of the ACO algorithm and of the genetic algorithm with adaptation and modification according to the subject area that is considered.

5. Combined algorithm for solving the problem on agreements execution planning

The main ACO of the algorithm involves implementation of the collective intelligence principle on the example of the ant colony. For the search for extreme values of the objective function, a definite number of agents (artificial ants), which construct the same number of possible solutions to the problem, is used on each iteration. Among these solutions, the part of the best ones by objective function is selected. The obtained information is stored in a generally accessible data bank and used by the agents on the following iterations independently of each other. Each agent operates under the rules of the probabilistic algorithm, and when choosing the direction, focuses not only on an increment of the objective function, but also on statistical information that reflects the previous history of a collective search [6].

The genetic algorithm uses the iterative approach to improvement of results. A search for the best solution in the vicinity of the given one takes place on each iteration. If such solution is found, it becomes current and a new iteration starts. It continues until increment of the objective function decreases almost to zero or the given number of iterations is performed. To increase the probability of finding a global optimum, the multiple experiments with different initial points are used, which significantly increases the search time [8].

It is important to note that when taking into account the features of a subject area, the application of purely classical algorithms in solving the problem of planning agreements execution is rather time consuming since it is necessary to conduct a number of modifications and adaptations. The use of such modifications can have a significant negative impact on the algorithm operation effectiveness, as well as on accuracy of the results.

Given all the above, to solve the problem, the own combined algorithm that is at once intended to be used for the considered subject area was proposed. The proposed algorithm will make it possible to take into account all features in a "natural" way, rather than as additional improvements and changes.

To use the combined algorithm, input data about the stages of executing agreements will be represented in the form of a multi-layer graph, in which one layer corresponds to execution of works by one executor. The nodes of the graph represent the intermediate state of transition from the implementation of one stage to another, and the edges that connect these nodes correspond directly to the process of implementation of stages. In this case, the nodes of each layer of the graph, which actually determine one moment of time, constitute the coherent whole, and therefore each node of each layer contains the data about implementation of all stages by executors. The weight of each edge is total duration of implementation of the correspondent stage. In addition, to predict the event of a delay of implementation of a certain stage, an additional edge between each pair of adjacent nodes was introduced in order to provide an opportunity of further transition in case it is impossible to implement any stage at a given moment.

The condition for obtaining the variant of the plan for implementation of agreement phases is visiting not all the nodes of the graph like in the case of using the classic ACO algorithm, but using all various edges, in addition to the edges, which correspond to a delay of each layer of the graph in order to form one complete route. Since the nodes correspond to moments of time, stopping at each of them not always means the need to proceed to the next node by another edge. Thus, there is an opportunity to keep consistency between the execution of works by various executors.

The combined algorithm for solving the problem of planning the execution of agreements contains the following main steps:

1. Determine the total number of agents that will be used to search for a solution from the formula (11):

$$b = m * \left((g-1) * \omega_n * \max \sum_{i=1}^{n} \sum_{j=1}^{J_i} y_{ijl} \right), \tag{11}$$

where m is the number of executors, involved in execution of agreement within the planned period; l is the number of an executor, $l \in M$, $M=\{1, 2,...m\}$ is the set of executors; g is the number of nodes of the graph; n is the number of agreements to be executed within the planned period; ω_n is the total number of stages for n agreements; i is the number of an agreement; J_i is the set of stages of the i-the agreement; j is the number of the stage from the set of stages, $j \in J_i$; y_{ijl} is the parameter that takes values $\{0,1\}$: $y_{ijl}=1$, if the j-th stage of the i-th agreement is executed by the l-th executor; $y_{ijl}=0$ in another case.

2. From formula (12), we determine the number of "elite" agents. "Elite" agents are the special agents that travel only along the best routes, found on the previous iterations.

$$b_l = \frac{b}{2*e}, \tag{12}$$

where $e \approx 2.718$ [6].

The speed of finding the optimal solution with the use of "elite" agents increases, since this way allows reduction of the number of iterations of the whole algorithm.

3. Set random permissible values of coefficients of the ACO algorithms, specifically:

α is the pheromone weight, 0≤α≤1, which determines relative significance of the influence of marks of the choice of a route, the shortest transition edge will be chosen at α=0, the edge, on which the level of marks will be the highest, will be chosen at α=1;

p is the coefficient of upgrading the marks level, 0≤*p*≤1, which determines its relative decrease in time;

Q is the parameter, the value of which has to be close to the length of the route that is optimal for current iteration;

A is the coefficient of "significance" of special agents, which determines the measure of strengthening the variant of the route.

4. Considering the ratio of the number of special agents to ordinary ones, choose an agent for the next iteration. If we choose an ordinary one – proceed to step 5, a special one – to step 11.

5. Check if all edges of the graph were passed. If so, proceed to step 9, if not – to step 6.

6. Check whether the implemented stage is completed by the given moment, which in turn implies the need to select the next edge for transition. If so, proceed to step 7, otherwise, – to step 5.

7. Add the data about the completed stage to the list of the completed stages to ensure the possibility of compliance with the requirements for the sequence of execution.

8. Check the possibility to complete the next stage, taking into account the requirements for the sequence of execution. If there is such possibility, we determine the next edge for transition by a modified probabilistic-proportional rule.

9. Mark the path that the agent traveled and clear the list of implemented stages, thus preparing the graph for the passage by the next agent.

10. Update the marks level at all edges, passed by an agent. For this, calculate the value of the generalized criterion F_0 for the current variant of the solution, which represents the route $T_k(t)$, which the k-th agent traveled at iteration t. The level of marks is determined from formula (13):

$$\Delta\tau_{ur,k}(t)=\begin{cases}\frac{1}{F_0}*\frac{Q}{L_k(t)}, & \text{if } (u,r)\in T_k(t),\\ 0, & \text{if } (u,r)\notin T_k(t),\end{cases} \qquad (13)$$

where u and r are the indices of pairs of nodes that connect the edge that the agent passed; $L_k(t)$ is the length of the path $T_k(t)$.

11. Strengthen the best found solutions using a special agent from formula (14):

$$\Delta\tau_{ur,k}(t)=A*\frac{1}{F_0}*\frac{Q}{L_k(t)}. \qquad (14)$$

12. Enumerate the values of ACO coefficients of the algorithm α, *p*, *Q*, *A* with the use of the genetic algorithm by the following scheme:

– based on current values, get the initial set of chromosomes;

– calculate the survival coefficient, specifically, put in correspondence to each solution or a chromosome a certain numeric value, dependent on proximity of this solution to the best variant by the value of the general estimation function, obtained by the given moment [1];

– perform displaying, in the process of which the chromosomes that have a high survival coefficient, and actually higher numeric value, are more likely to get to descendants, after which operators of mating and mutation are performed;

– form the next generation, the value of parameters of which will use the following set of agents in finding the path.

13. Compare current solutions with the ones, obtained on previous iterations and check them for optimality. If the found solutions are the best, or all ordinary and special agents have been used, proceed to step 14, otherwise – to step 4.

14. Derive the obtained results, which are the variant of the plan of agreement stage implementation. The obtained variant of the solution includes sequences of implementation of agreement stages in the planned period, marked on the time scale.

The proposed algorithm has been designed to be used when solving the problem on agreements execution planning. Its use enables finding solutions to the problem, taking into account all features. An experimental modeling of such algorithm showed that the used elements of the ACO algorithm and the elements of the genetic algorithm combined with representation of input data make it possible to find a solution that is close to the optimal one quite accurately.

We compared the proposed combined algorithm for searching and forming an optimal agreements execution plan with the classic and most heuristic algorithms. The results of the comparison of the proposed method allow us to considerably decrease the time for formation of an optimal plan while increasing the amount of input data. In comparison with the genetic algorithm, the proposed algorithm finds a more optimal solution, in addition, this search time was by 10–15 % shorter. It should be noted that most heuristic algorithms are related to approximate, so at the same input data, such algorithms can omit optimal values and give only approximate ones. During the experiment, most classical methods require by 60–80 % more time than the proposed one, which was especially noticeable when increasing the number of agreements – 50 or more. It is important to note that the result of the work depends not only on the number of agreements, but also the number of stages in all agreements.

However, it should be noted that application of each algorithm, including the proposed one, requires significant time consumption for preparation and processing of input data, as well as the application of information resources that will ensure information preparation and processing. In addition, depending on the conditions of execution of the agreement and its each stage, there arises the need to assign additional conditions and restrictions before applying the algorithm. A special feature of the proposed algorithm is that it takes into account the assigned mathematical model, which used additive criteria convolution. Additive convolution allows adjustment the relative significance of each criterion, which, in turn, makes it possible to take into account important aspects when finding an optimal plan depending on the general economic and social influences.

Given all the above, there is a need for designing and using a decision support system that will provide support in solving the problem of agreement execution planning with the possibility of taking account the general economic and social influences during the search for an optimal plan.

6. Design and construction of elements of a decision support system

To implement the proposed algorithm, there was created a programming module, which found its use in the development of the web-oriented DSS for agreement execution planning at the enterprises, the activity of which relates to rendering services. The purpose of the DSS is the implementation of the planning function, as well as support with necessary information. The system is designed for a wide range of users, which are proposed to be conditionally divided into the following groups:

– managers, responsible for meeting the conditions of agreements;

– heads of senior management;

– heads of production units;

– staff of production units and subcontractors;

– heads of departments of material and technical support;

– representatives of customers of services.

The functions for each of the following groups of users are represented in the form of the Use Case diagram (Fig. 1).

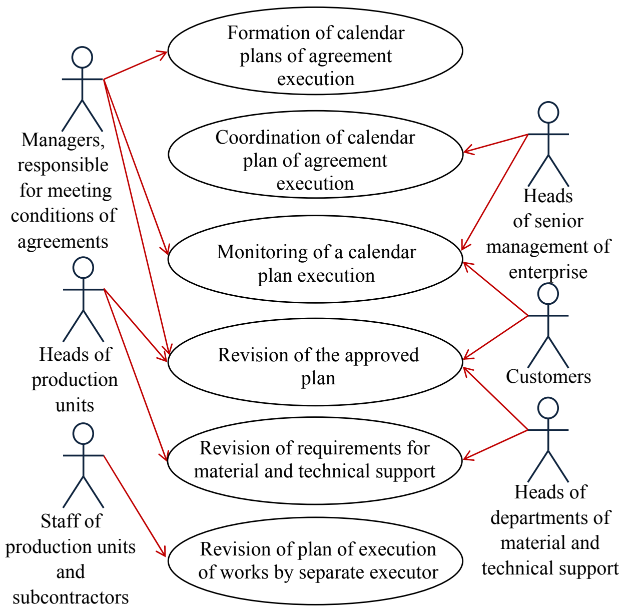

Managers, responsible for meeting conditions of agreements
Heads of production units
Staff of production units and subcontractors
Formation of calendar plans of agreement execution
Coordination of calendar plan of agreement execution
Monitoring of a calendar plan execution
Revision of the approved plan
Revision of requirements for material and technical support
Revision of plan of execution of works by separate executor
Heads of senior management of enterprise
Customers
Heads of departments of material and technical support

Fig 1. Use Case diagram of DSS use

When developing DSS for agreements execution planning, practical implementation of the client-server architecture in the form of a "thin" client was used – all logic was focused on the server part, and the client part was involved only in data display. In other words, the result of execution is displayed in a user's browser. The choice of this architecture of the construction of a system is caused by the following advantages:

– simplified integration with outside program products;

– implementation simplicity, which is explained by the absence of the need to adjust hardware or software tools of users;

– increased fault tolerance, because all software that is necessary for work is installed only on the server.

In addition, DSS is integrated with information systems, existing at an enterprise, in particular with the system of contractual relations management.

When designing the systems and the elements, all peculiarities of the main problem of planning were taken into account, as well as the need to provide all stages of planning with information support [12]. The main components of the system include (Fig. 2): ETL (Extract/Transform/Load) tools, Data Warehouse (DW), the server part, the web-based user's interface. It is important to note that the user interface is implemented as web-pages to be displayed in user browser.

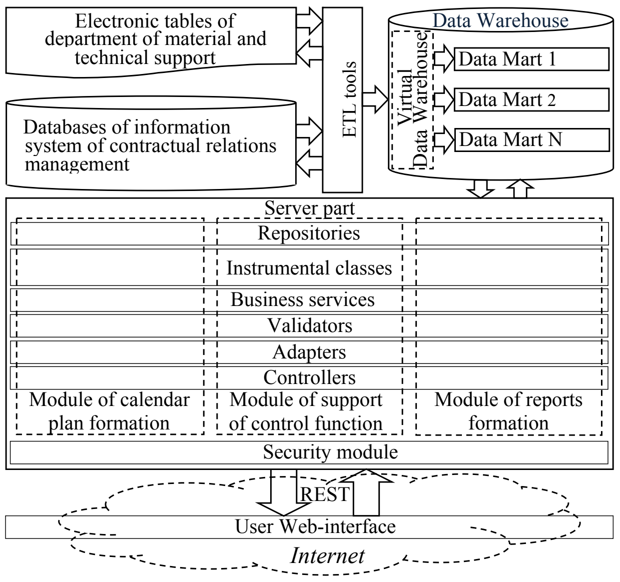

Electronic tables of department of material and technical support
Databases of information system of contractual relations management
ETL tools
Data Warehouse
Virtual Data Warehouse
Data Mart 1
Data Mart 2
Data Mart N
Server part
Repositories
Instrumental classes
Business services
Validators
Adapters
Controllers
Module of calendar plan formation
Module of support of control function
Module of reports formation
Security module
REST
User Web-interface
Internet

Fig. 2. DSS structure

A database of the information system of management of contractual relations and electronic tables of the department of material and technical support are data sources for DW. Close integration and connection that the created DSS with information carriers, existing at an enterprise, ensure adequacy and reliability of information support for managerial decision-making support while forming an agreement execution plan. Accumulation of information in DW goes on in the automatic mode by means of using the ETL tools, which is an intermediate link between the DSS and the existing information systems.

The classic multi-layer architecture with division into the following components was used when constructing the programming part:

– the layer of controllers, with the help of which interaction with the client part is provided;

– adapters, at the layer of which the data from clients' queries are transformed into basic business objects that are subsequently processed;

– classes that check input data for conformity with the set rules;

– business services that are directly responsible for business logic implementation;

– auxiliary instrumental classes, to which business services delegate execution of separate operations;

– repositories, meant for faultless work with DW.

The functional capabilities of the server part meet the needs of DSS users and in the general form are divided into the modules, designed to solve specific problems [13]. When processing a query, each module addresses the DW to obtain data. After data processing and necessary calculations, the result is displayed by the client part in the assigned form. The number of modules is not regulated, which provides, if necessary, upgrading of existing and inclusion of new modules to address the decision-making problems. The main modules of the system include the module of formation of calendar plan of agreements implementation, the module of support of function of execution control, the module of formation of reporting documentation.

The module of formation of the calendar plan of agreements implementation is intended to form the variants of the calendar plan using the proposed combined algorithm. The possibility to evaluate each variant, compare, make adjustments and determine additional parameters is also foreseen. The constructed and selected plan is transmitted in the electronic form to responsible parties for consideration and after approval is entered to databases of information systems that operate at an enterprise, in particular the system of management of contractual relations.

Module of support of execution of control function ensures provision of complete and relevant information on implementation of the agreement stages, compliance with the terms of a calendar plan, motion of resources, and materials, etc.

Module of formation of reporting documentation performs the functions of creation of various types of reporting documentation in the user-selected forms and representations.

Development of the program logic of operation of the above modules at the server part is performed in the Java programming language, the selection of which is justified by known advantages [14, 15], with the use of the Spring Framework and Hibernate. Even though the Java programming language provides a considerable number of built-in functionalities for development of information systems, the tools for incorporation of software components of a complex system in the integrity are not available. Solution of this problem rests directly on developers and architects. To solve this problem, the software platform Spring Framework, which provides formal means of combining disparate components into one programming system, based on the Dependency Injection principle, was selected [16]. In accordance with the mentioned principle, the problem of constructing dependencies of an object is solved by using an external mechanism. The key purpose of software platform Spring Framework is to ensure construction of the software infrastructure of the DSS.

Spring Framework has a modular structure, which allows using only the elements that are necessary for the implementation of functions of the system [17].

The modules are united into groups based on a functional purpose (Fig. 3), specifically:

– Core Container;

– Data Access/Integration modules;

– modules for development of Web-oriented systems;

– modules for working with AOP – Aspect Oriented Programming;

– instrumentation modules;

– test modules.

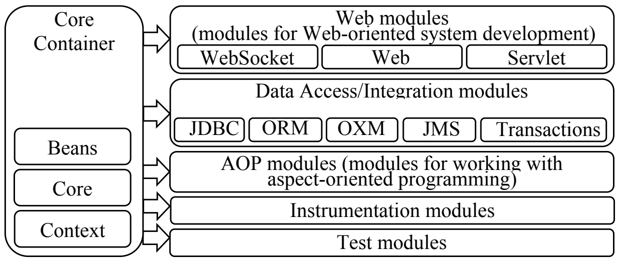

Core Container
Beans
Core
Context
Web modules (modules for Web-oriented system development)
WebSocket
Web
Servlet
Data Access/Integration modules
JDBC
ORM
OXM
JMS
Transactions
AOP modules (modules for working with aspect-oriented programming)
Instrumentation modules
Test modules

Fig. 3. Modules of software platform Spring Framework, divided into groups by a functional purpose

When developing DSS during agreement execution planning, the considered below modules were used.

A fundamental part of the software platform Spring Framework consists of modules of Beans and Core, which provide the possibility of introducing dependencies between the DSS components. This makes it possible to separate the configuration of the system and specification of objects' dependences. Context module expands Beans, provides a built-in support for the following functionality:

– localization and internationalization with the use of resource bundles;

– usage and distribution of events between components for behavior control;

– creation of program environments (contexts) for controlling the time of existence and interaction of components (for example, servlet container).

JDBC module provides its own abstraction layer for direct work with database and Data Warehouse, which provides a reduction of scope of work for developers, and also eliminates the need to process specific errors from specific database management systems. ORM module is the basis for the work with technologies for object-relational display, such as Hibernate. Module for messaging (JMS) provides functions to create and receive messages, which ensures interaction between the developed DSS with outside information systems. The Transactions module supports program and declarative transactions control, which provides the possibilities for flexible control of data integrity, used by the system, and unity of the operations that are performed [18].

Spring ensures the work with local and global transactions, without connection with implementation technologies. A sufficiently broad support for implementation of declarative transactions is provided both with the use of XML-configuration and with the use of annotations. In addition to this, there is the possibility to control transactions with a program, which creates flexibility at adjustment depending on specific requirements for the functionality of a system.

Modules for development of Web-oriented systems provide key opportunities to develop Web-services controllers for Web-integration of the system, which is crucial for interaction of the server part with the client one [19–21].

AOP module is responsible for working with aspect-oriented programming and provides the possibility to create methods-interceptors. The following methods are quite useful if it is necessary to separate the through line for different functionality objects. The possibilities of aspect-oriented programming are used for additional keeping record of information on all transactions that are carried out during the operation of the system with a description of all the calls and values of transmitted arguments. In addition, there is a possibility of integration with the AspectJ library, which provides extended ways to intercept calls and implement some program logic at various stages of interaction with objects.

Test modules, which provide an opportunity to test the components under control of Spring Framework, were used for performing integration testing of the system's components in conjunction with the unit testing library Junit. The software platform Mockito, which enabled creating mock objects for isolated checking the smallest fragments of the code in accordance with the principles of testing approach, was used for testing. Thus, iteration testing of the program code of the developed DSS is carried out using the data of modules and technologies.

Selection of a software platform for construction of the information system architecture is determined by the possibility of complete independence of business logic from the elements of a software platform and isolation of dependencies on components from the rest of the program code. This benefit is provided by Spring Framework, which also influenced its selection as a software platform for construction of software architecture.

When developing DSS, attention is paid to working with data sources, because interaction between the objects of the object-oriented programming languages and relational database is usually quite cumbersome and time consuming. This is caused by the paradigm gap between the way the object-represented data are stored compared with databases. To link databases with the concepts of object-oriented programming languages, it is appropriate to use ORM (Object-Relational Mapping) technology. The ORM concept in Java EE is represented by JPA (Java Persistence API) specification.

One of the implementations of the approach of object-relational mapping, specifically Hibernate, was used for working with DW. The task of Hibernate is the data transfer from the rows of database tables into the Java objects and vice versa [22]. In addition, Hibernate provides built-in features for queries and searches, thereby reducing the development time, spent on manual data processing in SQL and JDBC. However, unlike many similar solutions, Hibernate does not hide the capacity of SQL and if necessary, enables using relational technologies of the direction. The selection of Hibernate is also due to the lack of the need for complex interaction with the database, using complex queries etc., since data operation takes place at the program level. In comparison with the analogues (e. g., MyBatis), Hibernate immediately offers a ready solution, the initial requirements for configuration of which are minimal.

A general scheme of interaction of the level of access to the data of the information system with the relational database using Hibernate is shown in Fig. 4.

The interaction between the code of the information system and functional capabilities of Hibernate can take place through a standard interface of access to JPA, as well as through the extended interface Hibernate, which provides additional possibilities.

The interaction of the server part with the client's part is carried out by the Internet, using the REST (Representational State Transfer) ideology, which made it possible to separate the system's logic from the user interface and the make the system's structure simpler and more scalable [23]. REST is an architectural style for distributed systems. According to this style, each data element is unambiguously determined by a global identifier, such as a URL (Uniform Resource Locator).

For the formalized description of interfaces of interaction between the server and the client part, there was used specification Open API, through application of which all the resources, provided by the server, and accessible operations on them were described. Actually, a given specification determines the standard for describing REST interfaces, which allows both a person and a computer to understand the capabilities of the server without an access to the input code, additional technical documentation or direct interaction via the net [24].

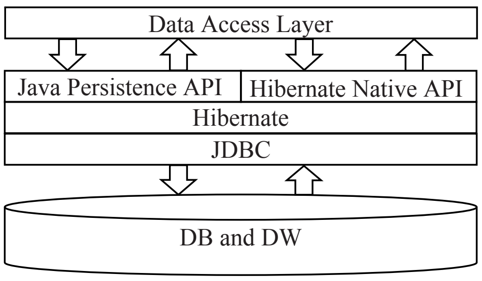
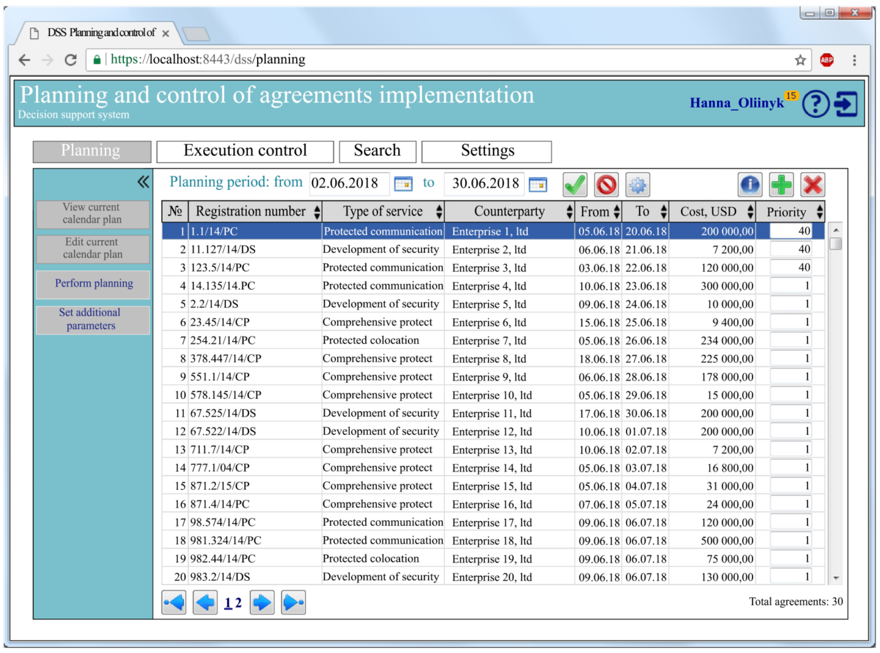


Fig. 4. Scheme of interaction of the Data Access Layer in the information system with the database and Data Warehouse with the use of Hibernate

The DSS programming code was developed using the approach of continuous integration (CI) and software product Jenkins. The use of CI is an approach to software development, in which each change in the program code is automatically fixed and tested [25]. Moreover, the actions such as compilation, downloading and updating the database scheme, and deployment on the server are performed automatically. Using CI reduces the risk of appearance of the problems that were not identified immediately, makes it easier to improve the program code, and performs the automation of trivial actions, enabling one to concentrate on the development of program logic of the system's functionality support.

Let us consider the basic capabilities of the developed web-oriented DSS to ensure compliance with the requirements regarding the necessary functionality for construction of plans of works execution and execution control based of two menu sections: "Planning" and "Control of execution".

The section "Planning" (Fig. 5) provides possibilities for reviewing or editing existing calendar plans, as well as planning execution of works for future periods.


DSS Planning and control of
https://localhost:8443/dss/planning
Planning and control of agreements implementation
Decision support system
Hanna_Oliinyk
15
Planning
Execution control
Search
Settings
View current calendar plan
Edit current calendar plan
Perform planning
Set additional parameters
Planning period: from
to
1 2

Planning period: from 02.06.2018 to 30.06.2018

№	Registration number	Type of service	Counterparty	From	To	Cost, USD	Priority
1	1.1/14/PC	Protected communication	Enterprise 1, ltd	05.06.18	20.06.18	200 000,00	40
2	11.127/14/DS	Development of security	Enterprise 2, ltd	06.06.18	21.06.18	7 200,00	40
3	123.5/14/PC	Protected communication	Enterprise 3, ltd	03.06.18	22.06.18	120 000,00	40
4	14.135/14.PC	Protected communication	Enterprise 4, ltd	10.06.18	23.06.18	300 000,00	1
5	2.2/14/DS	Development of security	Enterprise 5, ltd	09.06.18	24.06.18	10 000,00	1
6	23.45/14/CP	Comprehensive protect	Enterprise 6, ltd	15.06.18	25.06.18	9 400,00	1
7	254.21/14/PC	Protected colocation	Enterprise 7, ltd	05.06.18	26.06.18	234 000,00	1
8	378.447/14/CP	Comprehensive protect	Enterprise 8, ltd	18.06.18	27.06.18	225 000,00	1
9	551.1/14/CP	Comprehensive protect	Enterprise 9, ltd	06.06.18	28.06.18	178 000,00	1
10	578.145/14/CP	Comprehensive protect	Enterprise 10, ltd	05.06.18	29.06.18	15 000,00	1
11	67.525/14/DS	Development of security	Enterprise 11, ltd	17.06.18	30.06.18	200 000,00	1
12	67.522/14/DS	Development of security	Enterprise 12, ltd	10.06.18	01.07.18	200 000,00	1
13	711.7/14/CP	Comprehensive protect	Enterprise 13, ltd	10.06.18	02.07.18	7 200,00	1
14	777.1/04/CP	Comprehensive protect	Enterprise 14, ltd	05.06.18	03.07.18	16 800,00	1
15	871.2/15/CP	Comprehensive protect	Enterprise 15, ltd	05.06.18	04.07.18	31 000,00	1
16	871.4/14/PC	Comprehensive protect	Enterprise 16, ltd	07.06.18	05.07.18	24 000,00	1
17	98.574/14/PC	Protected communication	Enterprise 17, ltd	09.06.18	06.07.18	120 000,00	1
18	981.324/14/PC	Protected communication	Enterprise 18, ltd	09.06.18	06.07.18	500 000,00	1
19	982.44/14/PC	Protected colocation	Enterprise 19, ltd	09.06.18	06.07.18	75 000,00	1
20	983.2/14/DS	Development of security	Enterprise 20, ltd	09.06.18	06.07.18	130 000,00	1

Total agreements: 30

Fig. 5. Parameters of section "Planning"

After entering the initial and the final dates of the planned period, the list of agreements to be executed within the set period is formed. This list is presented in a tabular form with brief information about each agreement (number, registration number, type of a service, counterparty, term of validity, cost, and priority). For each agreement, there is a possibility to review detailed information, including the list of stages with executors and their sequence. The list of agreements can include other agreements or delete the existing ones. For each agreement, it is possible to set the implementation priority as a number from 1 to 100. By default, the set priority value is 1 (the smallest). When calculating the efficiency criterion, the entered priorities may be calculated from formula (15):

$$u_i = \frac{\upsilon_i}{\sum_i \upsilon_i}, \tag{15}$$

where υ_i is the priority value for each agreement, entered by a user.

Some variants of calendar plan can be formed when pushing the key "planning" (Fig. 6).

A visual representation of the execution variant is a table, in which executors are in rows, and the days of the planned period with indication of the day of the week are in columns. A table cell at the intersection of a line and a column shows completion of a stage of a certain agreement on a specified day. It displays the registration number of the agreement and the stage number and name.

Checking and improvement of certain parts of the plan take place at every stage. The person who makes up a plan tries to achieve the optimum agreement execution plan in the dialogue mode.

7. Discussion of results of DSS development and application

The improved mathematical model of agreement execution planning, the combined algorithm for solving problems of planning, the structure of the web-oriented DSS were proposed, and their use was substantiated.

The implemented functionality of the web-oriented DSS provides the solution to the basic problem of decision-making. In addition, web-orientation of the developed system enables the formation of a calendar plan by authorized representatives of an enterprise in any place, where there is the Internet network. This allows adjustment of the agreement execution plan in the operative mode with minimization of time and costs. In this case, it is possible to adjust the plan immediately after negotiations on the side of a customer and evaluate available technical possibilities.

Effectiveness of web-oriented DSS in planning and controlling agreement execution is determined by the following factors:

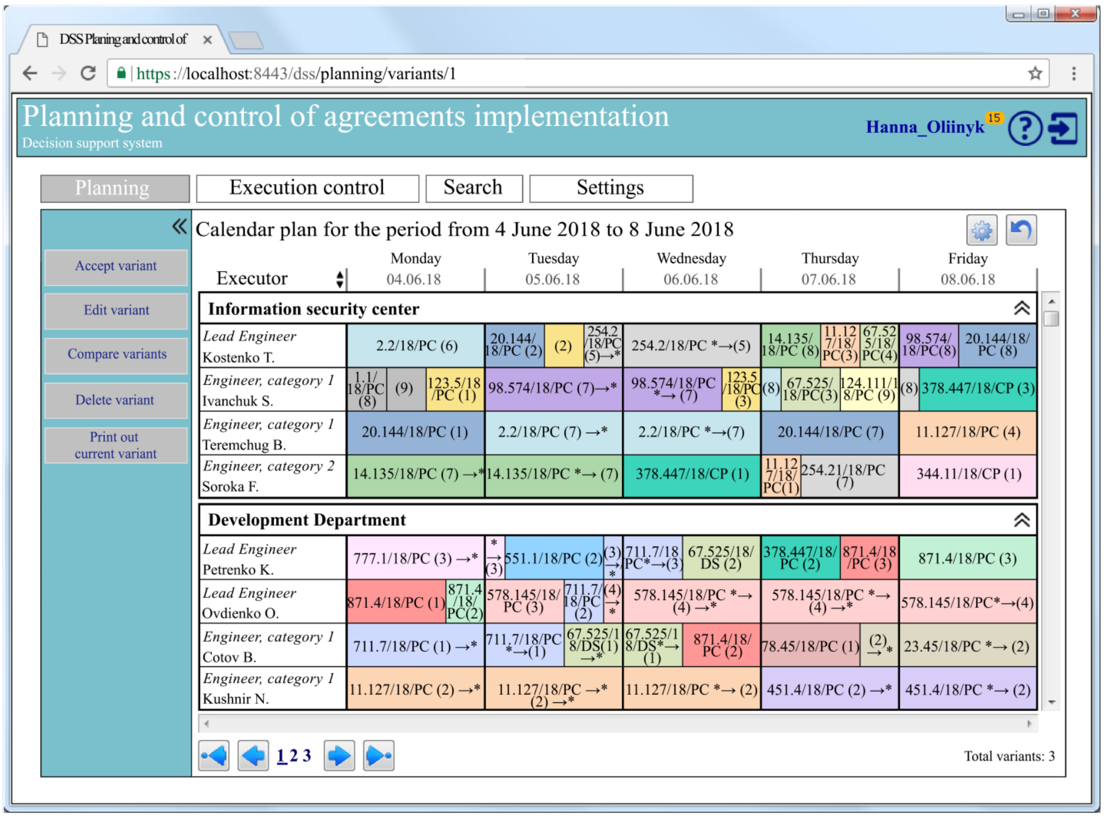
– operative formation and flexibility at changing a calendar plan that accelerates the execution of customers' orders and reduces the amount of downtime during implementation;

– significant reduction of time consumption for data collection and information preparation;

– possibility of quick adaptation to the current situation and making corresponding changes to a calendar plan;

– distinct division of time of works between executors, which takes into account the sequence of execution and individual characteristics of each agreement;

– possibility of permanent control of the state of execution of works and evaluation of the execution of a separate agreement.


DSS Planing and control of
https://localhost:8443/dss/planning/variants/1
Planning and control of agreements implementation
Decision support system
Hanna_Oliinyk 15
Planning
Execution control
Search
Settings
Accept variant
Edit variant
Compare variants
Delete variant
Print out current variant
1 2 3

Calendar plan for the period from 4 June 2018 to 8 June 2018

Executor	Monday 04.06.18	Tuesday 05.06.18	Wednesday 06.06.18	Thursday 07.06.18	Friday 08.06.18
Information security center					
Lead Engineer Kostenko T.	2.2/18/PC (6)	20.144/18/PC (2); (2); 254.2/18/PC (5)→*	254.2/18/PC *→(5)	14.135/18/PC (8); 11.127/18/PC(3); 67.525/18/PC(4)	98.574/18/PC(8); 20.144/18/PC (8)
Engineer, category 1 Ivanchuk S.	1.1/18/PC (8); (9); 123.5/18/PC (1)	98.574/18/PC (7)→*	98.574/18/PC *→ (7); 123.5/18/PC (3)	(8); 67.525/18/PC(3); 124.111/18/PC (9)	(8); 378.447/18/CP (3)
Engineer, category 1 Teremchug B.	20.144/18/PC (1)	2.2/18/PC (7) →*	2.2/18/PC *→(7)	20.144/18/PC (7)	11.127/18/PC (4)
Engineer, category 2 Soroka F.	14.135/18/PC (7) →*	14.135/18/PC *→ (7)	378.447/18/CP (1)	11.127/18/PC(1); 254.21/18/PC (7)	344.11/18/CP (1)
Development Department					
Lead Engineer Petrenko K.	777.1/18/PC (3) →*	*→(3); 551.1/18/PC (2); (3)→*	711.7/18/PC*→(3); 67.525/18/DS (2)	378.447/18/PC (2); 871.4/18/PC (3)	871.4/18/PC (3)
Lead Engineer Ovdienko O.	871.4/18/PC (1); 871.4/18/PC(2)	578.145/18/PC (3); 711.7/18/PC (2); (4)→*	578.145/18/PC *→ (4) →*	578.145/18/PC *→ (4) →*	578.145/18/PC*→(4)
Engineer, category 1 Cotov B.	711.7/18/PC (1) →*	711.7/18/PC *→(1); 67.525/18/DS(1) →*	67.525/18/DS*→ (1); 871.4/18/PC (2)	78.45/18/PC (1); (2)→*	23.45/18/PC *→ (2)
Engineer, category 1 Kushnir N.	11.127/18/PC (2) →*	11.127/18/PC →* (2) →*	11.127/18/PC *→ (2)	451.4/18/PC (2) →*	451.4/18/PC *→ (2)

Total variants: 3

Fig. 6. Created variant of the calendar plan

The architecture of the created web-oriented DSS involves minimal costs of its implementation at the related enterprises. Due to modularity, the system easily adapts and integrates with information systems, existing at an enterprise. To access the capabilities of the system by a user, it is necessary to have an installed web-browser and an access to the Internet network.

Without the use of the created web-oriented DSS, it is not possible to carry out practical implementation of the combined algorithm. It is the DSS that provides data preparation and carrying out the necessary calculations and makes it possible to choose variants and make corrective changes.

The benefits of applying the developed DSS at an enterprise imply the reduction of time required to process available data, in particular, if necessary, adjustment of the existing calendar plan. Formation of the production program and plans for the system implementation typically includes the following steps:

– setting the planned period;

– analysis of the works schedule and selection of the executors who are not engaged during these periods;

– analysis and determining parameters of completing each stage of each agreement;

– formation of variants of the production program and plans;

– assessment of the formed variants of the production program;

– making decisions about selection of the variant of the production program and plans;

– adjustment of input parameters, if necessary, which is the ground for repeated formation of solution variants.

Using the developed DSS, formation of execution variants is imposed on the system. In this case, there is the possibility of restructuring variants, taking into account actual changes, when making some adjustments. Overall, due to the automation of formation and adjustment of the calendar plan variants and displaying the existing state of work execution, the promptness and timeliness of rendering services to clients is ensured.

Further development of the research and development of the system is aimed at the expansion of functional possibilities for intellectual data analysis and processing large data arrays. Using the methods for intellectual data analysis will provide information analysis and detection of regularities and factors that may affect effectiveness of agreements' execution. In addition, detection of such regularities will provide an additional possibility of forming execution patterns that are advisable to use in order to reduce the time of the combined algorithm. It is important to note that during agreements execution planning the DSS makes it possible to concentrate on analysis, assessment and selection of the optimum variant of a solution, performing all preparatory actions and forming ready variants of solutions.

8. Conclusions

1. The mathematical model of agreements execution planning was developed with respect to the main features of activity of service-rendering enterprises, which takes into account the main partial criteria and limitations. The use of the additive convolution of criteria allows the adjustment of relative significance of each criterion, which in turn makes it possible to take into consideration important aspects when finding the optimal plan depending on general economic and social influences.

2. The combination of elements of the ACO algorithm and of the genetic algorithm became the basis for the proposed combined algorithm for the search and formation of the optimal agreements execution plan, which allows us to reduce time for the formation of an optimal plan while increasing input data in comparison with the classic and most heuristic algorithms.

3. We proposed the structure of the web-oriented DSS, which provides a complete solution to the problem on agreements execution planning as it allows focusing on analysis, evaluation and selection of the best variant by performing preparatory actions and forming ready variants of solutions. In addition, in the course of development of the system its integration into existing information systems and minimal costs for the DSS implementation were taken into consideration. It also becomes possible to adapt easily the use of the main modules of the system to solve the related problems.

4. The use of the technologies and tools for construction of the DSS components was substantiated. The results of development and operation of the created DSS were represented, which made it possible to evaluate the benefits of using it during testing and to make sure that efficiency of the planning process increased. Due to the DSS use, it became possible to offer the customers the proposals regarding the time limits of the agreement execution, to significantly reduce the time required to construct the variants of works execution, evaluation, and selection. Previously, that took more than one entire working day.

References

1. Hrybkov S. V., Lytvynov V. A., Oliinyk H. V. Zadacha planuvannia vykonannia dohovoriv ta pidkhody do yii efektyvnoho vyrishennia // Matematicheskie mashiny i sistemy. 2015. Issue 2. P. 61–70.
2. Bakshi T., Sarkar B., Sanyal S. K. An Evolutionary Algorithm for Multi-criteria Resource Constrained Project Scheduling Problem based On PSO // Procedia Technology. 2012. Vol. 6. P. 231–238. doi: 10.1016/j.protcy.2012.10.028
3. Chastikova V. A., Vlasov K. A. Razrabotka i sravnitel'nyy analiz evristicheskih algoritmov dlya poiska naimen'shego gamil'tonova cikla v polnom grafe // Fundamental'nye issledovaniya. 2013. Issue 10. P. 63–67.
4. Santosh K. S., Vinod K. G. Genetic Algorithms: Basic Concepts and Real World Applications // International Journal of Electrical, Electronics and Computer Systems (IJEECS). 2015. Vol. 3, Issue 12. P. 116–123.
5. A Hybrid Algorithm Based on PSO and ACO Approach for Solving Combinatorial Fuzzy Unrelated Parallel Machine Scheduling Problem / Senthilkumar K. M., Selladurai V., Raja K., Thirunavukkarasu V. // European Journal of Scientific Research. 2011. Vol. 64, Issue 2. P. 293.

6. Hrybkov S. V., Oliinyk H. V. Modyfikovanyi ACO alhorytm pobudovy kalendarnoho planu vykonannia dohovoriv // Matematychne ta kompiuterne modeliuvannia. Seriya: Tekhnichni nauky. 2017. Issue 15. P. 156–162.
7. Emine A., Tugba S. A Variable Capacity Parallel Machine Scheduling Problem // Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management. 2012. P. 548–554.
8. An artificial bee colony algorithm for the maximally diverse grouping problem / Rodriguez F. J., Lozano M., García-Martínez C., González-Barrera J. D. // Information Sciences. 2013. Vol. 230. P. 183–196. doi: 10.1016/j.ins.2012.12.020
9. Sivaraj R., Ravichandran T., Devi Priya R. Boosting Performance of genetic algorithm through Selective initialization // European Journal of Scientific Research. 2012. Vol. 68, Issue 1. P. 93–100.
10. Shafiei Nikabadi M., Naderi R. A hybrid algorithm for unrelated parallel machines scheduling // International Journal of Industrial Engineering Computations. 2016. P. 681–702. doi: 10.5267/j.ijiec.2016.2.004
11. Algorithms and Methods for Solving Scheduling Problems and Other Extremum Problems on Large-Scale Graphs / Pankratiev E. V., Chepovskiy A. M., Cherepanov E. A., Chernyshev S. V. // Journal of Mathematical Sciences. 2005. Vol. 128, Issue 6. P. 3487–3495. doi: 10.1007/s10958-005-0283-z
12. Hrybkov S., Oliinyk H. Modeling of the decision support system structure in the planning and controlling of contracts implementation // Ukrainian Journal of Food Science. 2015. Vol. 3, Issue 1. P. 123–130.
13. Bass L., Clements P., Kazman R. Software Architecture in Practice. Addison-Wesley Professional, 2013. 640 p.
14. Niemeyer P., Leuck P. Learning Java. 4th ed. O'Reilly Media, 2013. 1010 p.
15. Ritter S. 4 Reasons Why Java is Still #1 // Azul Systems. 2018. URL: https://www.azul.com/4-reasons-java-still-1/
16. Walls C. Spring in Action. 5th Edition. Manning publication, 2018. 500 p.
17. Spring Framework. Reference Documentation / Johnson R., Hoeller J. et. al. // 2016. URL: https://docs.spring.io/spring/docs/4.3.9.RELEASE/spring-framework-reference/html/
18. Oliinyk H. V., Hrybkov S. V. Pidtrymka mekhanizmu tranzaktsiyi prohramnoiu platformoiu Spring // Mizhnarodnoi naukovo-tekhnichnoi konferentsiyi «Suchasni metody, informatsiyne, prohramne ta tekhnichne zabezpechennia system upravlinnia orhanizatsiino-tekhnichnymy ta tekhnolohichnymy kompleksamy». Kyiv: NUKhT, 2016. P. 250–251.
19. Moskalenko N. V. Web-orientirovannaya informacionnaya sistema reklamnoy kompanii // Ekonomicheskie nauki. 2017. Issue 61-1. URL: https://novainfo.ru/article/?nid=11602
20. Ahaev A. V. WEB-orientirovannaya ekspertnaya sistema vybora programmnyh produktov // Nauka. Tekhnologii. Innovacii: materialy Vseros. nauch. konf. studentov, aspirantov i molodyh uchenyh. Novosibirsk. 2012. P. 263–266.
21. Loboda Yu. G., Orlova O. Yu. Tekhnologii razrabotki Web-prilozheniy // Naukovi pratsi. Odesa: Odeska natsionalna akademiya kharchovykh tekhnolohiy. 2014. Issue 46. P. 239–244.
22. Hibernate ORM 5.2.13.Final User Guide / Mihalcea M., Ebersole S. et. al. // URL: https://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html
23. Kalin M. Java Web Services: Up and Running. 2nd ed. O'Reilly Media, 2013. 360 p.
24. OpenAPI Specification. URL: https://swagger.io/specification/
25. Smart J. Jenkins: The Definitive Guide. O'Reilly Media, 2011. 404 p.