

The parallel approach to issue of operational management optimization problem on transport gateway system

Alexander Egorov¹, Natalya Krupenina¹, and Lyubov Tyndykar^{1,}*

¹Admiral Makarov State University of Maritime and Inland Shipping, 5/7, Dvinskaya str, 198035 Saint-Petersburg, Russia

Abstract. The universal parallelization software shell for joint data processing, implemented in combination with a distributed computing system, is considered. The research purpose – to find the most effective solution for the navigable canal management information system organizing. One optimization option is to increase computer devices computing power by combining them into a single computing cluster. The management optimizing task of a locked shipping channel for execution to adapt in a multi-threaded environment is proposed with constraints on a technologically feasible schedule. In article shows algorithms and gives recommendations for their application in the subtasks formation in parallel processing case, as well as on a separate thread. The proposed approach to building a tree of options allows you to optimally distribute the load between all resources multi-threaded system any structure.

1 Introduction

Russian Federation inland waterways are unified transport system part that includes many shipping locks, some of these are integrated into lock shipping systems. To reduce operating and transportation costs, energy and time resources should be optimized. Along with this, to unload channels need to increase their throughput. The optimal solution is achieved through automation [1-4], that allows for the ship's pass operational management [5-12]. In [13] presented the current state of a simulation tool dedicated to the design of control elaborate methods from the automatic control domain. In work [14] the global management of the canal reach is discussed. It is worth noting that, the modern information technologies use has a huge role in the shipping management. Such technologies can be divided into automated process control systems, and automatic identification systems.

However, holistic implement automated dispatching systems at shipping channel have not been created. Given this, necessary to use systems for provide automatic exchange of information between ships and coastal services, as well as high-performance computing complexes.

It is proposed to consider a universal parallelization software shell for joint data processing, implemented in combination with a distributed computing system.

* Corresponding author: tyndykarln@gumrf.ru

The software shell main purpose is to organize the effective application solution implementation, using for this purpose the computing power of parallelization devices. In optimizing the gateway system management issues, combining them into a single computing cluster gives the following advantages:

- allows to form hardware and software systems of any computing power, and if necessary to increase it during operation;
- increases the fault tolerance of the system (with the objects functioning at this kind this is an important factor).

However, when using the parallelization software shell, taking into account computer performance limitations [15], to increase the calculations efficiency possible by combining them into a single distributed computing system only [16-17], possibly in the cloud [18]. Note that the parallel data processing implementation on the one computer devices and within the distributed computing system framework of are conceptually different tasks [19-20].

2 Materials and Methods

Consider the optimizing a navigable canal control problem and adapt the algorithm for execution in a multi-threaded environment.

As part of the task, the navigable channel consists of N facilities that perform ship maintenance. One of the directions along the channel is direct, in accordance with this the objects are as follows: 1, 2, 3, ..., N . Moreover, denote the planning period (1):

$$[t_0, t_0+T] \quad (1)$$

where t_0 – the beginning of planning period, a T – the planning horizon.

It is known that at the moment t_0 there are many ships forward $P_n(t_0)$ and reverse $P_0(t_0)$ directions on the navigable channel, and during the planning period $[t_0, t_0+T]$, many ships of forward $\Delta P_n^+(t_0, T)$ and reverse $\Delta P_0^+(t_0, T)$ directions will enter the channel.

Let $K_n(t_0)$, $K_0(t_0)$, $\Delta K_n^+(t_0, T)$ and $\Delta K_0^+(t_0, T)$ – the number of groups simultaneously locking ships for the sets $P_n(t_0)$, $P_0(t_0)$, $\Delta P_n^+(t_0, T)$ and $\Delta P_0^+(t_0, T)$. Then the total number of ships on the channel for the planning period is determined (2):

$$M = K_n(t_0) + K_0(t_0) + \Delta K_n^+(t_0, T) + \Delta K_0^+(t_0, T) \quad (2)$$

Will number the entire ship groups, given that $i = 1, 2, \dots, M$ (3):

$$\{P_n(t_0), \Delta P_n^+(t_0, T), P_0(t_0), \Delta P_0^+(t_0, T)\} = \{P_i\}. \quad (3)$$

The schedule R of locked shipping channel functioning for the planning period $[t_0, t_0+T]$ is the dimension $M*N$ table, in which each element (i,j) – the set $(\tau_{ij}^H, \tau_{ij}^K)$, where, τ_{ij}^H – the start time, a τ_{ij}^K – the technological operation end time at object j over i -th ships group.

A technologically feasible schedule must satisfy the following restrictions:

- restrictions to follow;
- restrictions on the tinning continuity;
- restrictions on the serving device settings;
- restrictions on schedule connectivity;
- limitations of the ship pass technological process.

Consider the restriction of follow. A lock navigable channel is a conveyor system with the countercurrent presence, the ships movement along which is stipulated by a strict channel objects passage sequence. This can be expressed by the following requirements:

- positive technological operation duration (4)

$$\forall i = 1, 2, \dots, M, \forall j = 1, 2, \dots, N : \tau_{ij}^K > \tau_{ij}^H; \quad (4)$$

- the subsequent operation beginning on the application cannot occur before the previous operation on it completion (5, 6)

$$\forall i = 1, 2, \dots, K_n(t_0) + \Delta K_n^+(t_0, T), \forall j = 1, 2, \dots, N-1 : \tau_{ij}^k < \tau_{ij+1}^h; \quad (5)$$

$$\forall i = K_n(t_0) + \Delta K_n^+(t_0, T) + 1, \dots, M, \forall j = 2, 3, \dots, N : \tau_{ij}^k < \tau_{ij-1}^h. \quad (6)$$

The servicing continuity limitation is expressed in the fact that the servicing object cannot start the next request execution until it has finished servicing the previous one (7):

$$\tau_{ik}^h < \tau_{jk}^h; \tau_{ik}^k < \tau_{jk}^k, \quad (7)$$

where $\forall i, j = 1, 2, \dots, M, \forall k = 1, 2, \dots, N$.

If the gateway performs sequential one flow requests processing, then between the previous end and the next service beginning it needs time for readjustment. This is the service apparatus setting limitation (8, 9):

$$\tau_{i+j}^h - \tau_{ij}^k > d_j^h, \quad (8)$$

where $\forall i = 1, 2, \dots, K_n(t_0) + \Delta K_n^+(t_0, T) - 1, \forall j = 1, 2, \dots, N$,

$$\tau_{i+j}^h - \tau_{ij}^k > d_j^0 \quad (9)$$

where $\forall i = K_n(t_0) + \Delta K_n^+(t_0, T) + 1, \dots, M - 1, \forall j = 1, 2, \dots, N, d_j^h, d_j^0$ – time of idle lock on the j -th gateway forward and reverse directions.

The restriction on schedule connectivity is applicable if the application is served by objects k and m . In this case, it should be serviced by all objects located on the request route between them (10):

$$\forall r: k < r < m (i, r) \in U \quad (10)$$

where $\forall i = 1, 2, \dots, M$, если $\exists k, m = 1, 2, \dots, N: (i, k) \in U, (i, m) \in U$.

The limitations of technological process ship passage are specific restrictions imposed by local navigation conditions in the shipping channels area.

2.1 Optimization problem statement

The constructing task an optimal schedule for a locked shipping channel is formulated as follows. It is required for the planning period $[t_0, t_0 + T]$ to form such a schedule R that satisfies all the described limitations and provides the best value for the optimization criterion, which can be economic, technological and production and economic indicators.

Economic indicator – the maximum income at fixed or differentiated rates for the channel passage. There can be several technological indicators:

- minimum amount of weighted channel time;
- minimum of the longest passage channel time;
- minimum amount of weighted deviations from the policy period;
- minimum average channel travel time.

Production and economic indicators are calculated in tonnage-days, as an example:

- minimum amount of weighted costs for the channel passage;
- minimum of the largest costs for the channel passage;
- minimum amount of weighted costs for deviations from the policy period;
- minimum average cost for the channel passage.

The optimizing the navigable channel operational management task can be formulated as an integer programming problem, the dimension of which does not allow it to be solved by exact methods using existing computer equipment. Therefore, a quasi-optimizing method for constructing a schedule based on the branch and bound method is proposed. The

its work principle is to simulate the shipping channel system operation during the planning period $[t_0, t_0+T]$, forming, at the same time, an all possible control options tree at a given interval, among which the best control actions are selected according to given criteria. For this, an information system for managing a navigable canal is used, which contains a simulation functioning model that generates valid control options and simulates the system behavior when these options are adopted.

It is worth noting, that real-time coupling of a hydro-environmental model and an optimization model is difficult because hydrodynamic model and scalar transport model are both time-consuming, especially when the computational domain is large [21-22].

2.2 The navigable canal management information system operation algorithm

The information system for managing a shipping channel operation is a multi-step recursive process. At each processing cycle, a certain gateway system state S at time t is considered, which includes such channel characteristics as the fleet located deployment on the channel during the planning period and the status of objects performing ship maintenance. At the cycle end, a new system states set is formed at a time $t' > t$, in which each corresponds to one of the many control options currently available. Then, in turn, for all new states, the same processing cycle is performed. As a result, step-by-step promotion and formation of a tree of options from the initial state of the system at a time t_0 until the planning period end t_0+T . In simulating the operation shipping channel process all events, including decisions made, are recorded and, based on them, for a better control effect, a schedule for the vessels is formed.

The algorithm of the system for managing a shipping channel information operation is presented in Table 1.

Table 1. The information system algorithm.

№	Description	Commentary
1	System simulation.	The transition from the initial state to the state of the moment in time at which it is necessary to make a control decision is considered.
2	Assessment of the schedule stability for the state.	If the value does not satisfy the required level, then this branch is not considered in the future and there is a return to the last branch point.
3	The construction of many possible control options.	Variants are built for the state at the moment. If there are more than one options, then a branch point will form on the tree.
4	Management consideration.	Only one control option is considered.
5	The algorithm is performed anew.	The remaining control options are remembered until the algorithm returns to this branch point.
6	Evaluation by the optimization criterion (the guiding effect in this case is presented as a decisions made set when moving from the initial system state to the current one).	If the planning period end is reached, then the planning horizon of the formed control action is taken into account. If it is better than the current optimal control, then become one.
7	Return to the previous branch point, where the next control action of this point is considered.	In unreasonable alternatives absence at a given point, the algorithm goes to an earlier branch point before reaching the root node.
8	Schedule creation corresponding to the stored optimal control.	Occurs after checking all possible control options.

Note that options tree dimension is influenced by:

- the number of ships and facilities involved in ship passing process;
- the depth of the forecast schedule.

Parallelization of the operation algorithm will allow, within the acceptable time limits, to increase channel area size and the planning period. This will have a positive effect on information system performance for managing navigable shipping channel, since forecast schedule quality will be improved.

2.3 Options tree formation

Formally, the tree structure is a finite set L consisting of one or more nodes such that there is one specially designated node called the tree root, and the remaining nodes, excluding the root, are contained in $m, m \geq 1$ pairwise disjoint sets L_1, L_2, \dots, L_m , each is a tree. Therefore, when parallelizing the shipping channel control system operation algorithm, the optimal control in one of the pairwise disjoint sets (subtrees) at the main variant tree can be considered.

To construct the indicated subtree, only its root is needed, during the processing at which all other bonds will be formed. Therefore, a parallel algorithm for the navigable canal information management system operation can be represented as follows:

- partial control options tree generation;
- sending the options resulting tree final nodes on a separate execution thread to continue simulation and search for the optimum in the corresponding subtree;
- best solution selection to the subtask from the received set.

The selected solution is the optimal control action within the entire options tree framework.

The optimal subtasks size is determined, of course, by the initial task volume and the technical multithreaded system characteristics. For example, threads may have restrictions on the solved subtask volume, which is more related to the memory available size to it. In many ways, this problem is solved at the design stage or the multi-threaded system selection to solve the problem. Also, the system can have both a homogeneous structure, the flows have the same performance, and heterogeneous. In the first case, to ensure maximum performance, it is necessary to form equal dimension subtasks, and their number should be a threads number multiple in the system in order to maximize the involvement for processing. In the second, it is necessary to introduce a devices operating productivity estimate in a computer system, on the basis optimal volume subproblem is formed for a particular stream. A property such as system homogeneity or heterogeneity also has an effect. In addition, the parallelization software shell supports simultaneous operation with a variety of various types hardware.

To Amdahl's, the execution parallel algorithm time cannot be less than the execution its longest fragment time. At the same time, it is also necessary to take into account the memory costs for storing the generated options tree.

Let the hardware performance assessment determine the same size subtrees number that devices can process in an equal time amount. If, taking into account cutoffs, the average branches number for each tree node is taken as G , then the nodes number in the tree is defined as $\sum G_d$, where $d=0, \dots, D$ is the tree depth level, and D is the depth tree levels number. Then, in the general case, in order for two subtrees from the same tree to have an equal nodes number, it is necessary that their roots be located at the same depth level.

To ensure this condition, an approach is proposed in which a tree is generated until end nodes reaches number a predetermined value K , which is calculated based on performance estimates set for all devices in the computing environment. This allows you to optimally distribute the load between all the resources of any structure multi-threaded system.

Note that parallel algorithm execution time cannot be less than its longest fragment execution time. Therefore, when parallelizing optimizing the ship's pass problem for the number K and evaluating the productivity of the devices, it is necessary to choose in such a way that the following conditions are met:

- for each stream, the calculation time was the same;
- the ratio of subtasks generation time and the search for optimal control in them was optimal each.

In addition, it is necessary to take into account the memory costs of variants storing the generated tree.

If we take the hardware performance estimate as the number same size subtrees (which the devices can process for an equal time interval) and denote the cutoff of branches average number for each tree node, it is determined by the tree depth level and tree depth levels number. At the same time, tree roots location at the same depth level provides an equal their subtrees nodes number.

In the considered sequential information management system algorithm, a node is considered located at the next tree depth level (the exception is those cases in which it is necessary to return to the previous branch point). As this progression result, the final tree nodes are located at various depth levels. Such movement in a tree is called vertical (Fig. 1). However, to generate a tree of subtasks during parallelization, it is required that its finished nodes be at equal depth levels. To ensure this condition, it is necessary to implement horizontal movement (Fig. 2) along the tree, in which successive passage of each current depth level node is performed, after which the transition to the next level is carried out. As a result, at any moment of time, all the final tree nodes are located at one or adjacent depth levels.

Navigable canal management information system operation algorithm in the horizontal passage nodes mode differs from the vertical only in that after the many formation possible control options $\{U_i\}$ for the state S' at a time t' , the options are not considered immediately, but are saved at the next tree level. At each new processing cycle, the next node in turn located at the same depth level, or the next level first node is considered, if all the current level states are already processed. It should be noted that with this movement nature, the transition to the previous branch point is not required.

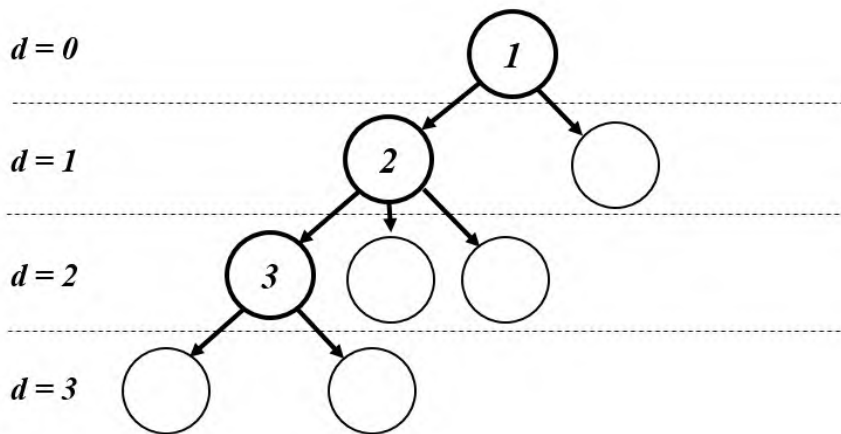


Fig. 1. Tree structure after three processing cycles with vertical movement.

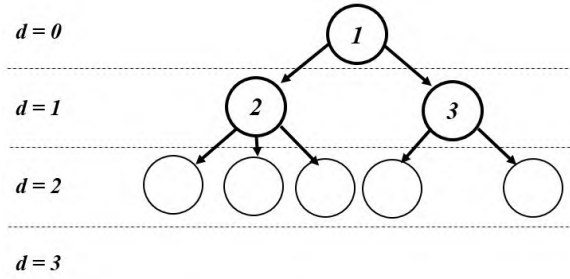


Fig. 2. Tree structure after three processing cycles with horizontal movement nature.

Considering that all the main stages of navigable canal management information system operation algorithm are preserved, its operation speed with both vertical and horizontal movement nature can be considered equal.

The main horizontal promotion disadvantage is that when reaching the planning period end at a certain level d , which limits the control options there, it is necessary to store data on G_d nodes in memory, in comparison with $G*d$ the vertical one. From it follows that the control system algorithm in the nodes vertical passage mode, ceteris paribus, consumes significantly less memory than horizontal.

2.4 Parallel operation algorithm

To describe the navigable canal management information system operation parallel algorithm introduce the following atomic statements (Table 2) corresponding to the main program's calculations events.

Table 2. Atomic statements used in the navigable canal management information system algorithm description.

Atomic statements	Description
add_taski	The finding i -th task optimal schedule from the initial state S at a time point t_0 to the depth T .
create_treei	Partial control options containing tree formation the required subtrees number K for the i -th task (implemented by executing the gateway control system algorithm in the mode of horizontal nodes passage until the end nodes number the tree is as close as possible to the number K , but will not exceed it).
all_senti	All end generated control options tree nodes at the i -th task are sent to the stream for further processing.
freei	i -th flow is free
busyi	i -th flow is busy
sendi	Sending the i -th end node to a separate execution thread to search for the corresponding subtree optimum.
solvei	The i -th subtree optimal control is found (for this, the navigable canal management information system algorithm in the vertical nodes passage mode is used).
recvi	The stream returns the found i -th subtree optimal control
save_best_solutionij	Obtained i -th subtree optimal contro comparison with the current j -th problem optimum and, if the new result is better than the current, then replaces it.
all_solvei	The all optimal controls i -th task formed subtrees are found and processed.
recv_timetablei	Returning the i -th task optimal schedule.

2.5 Temporal logic

Consider the temporal logic used to formulate a parallel algorithm mathematical model for navigable canal management information system operation:

- whenever a new task arrives, a control options tree is formed for it until the subtrees number in it reaches the required value;
- for each formed the i -th task subtree, sometime optimal controls must be found and processed;
- until i -th task all generated subtrees optimal controls are found and processed, unsent subtrees will be sent at free flows to search for the optimum in them, and the found optima will be received from the flows;
- whenever a subtree is sent to free flow, it becomes busy;
- whenever a thread becomes busy sometime optimal control should be found for the subtree it processes;
- after finding the optimal control, the flow returns it and becomes free;
- whenever the stream returns the corresponding subtree optimal control, it is compared with the current optimum (if the new result is better than the current one, then it is replaced);
- after the optimal controls the formed subtrees have been found and processed, a schedule is generated corresponding to the stored optimal control;
- after the optimal schedule formation, it is returned to the user.

The obtained temporal logic dependences allow us to present the model (Fig. 3, 4) main processes. A counter is set for each pseudo-code operation.

In accordance with the constructed pseudo-code, a marked-up transition system is built, where all possible navigable canal management information system operation states are taken into account, the main are:

- command execution counters;
- solving the problem (solved / not solved) state;
- flow state (busy / free).

```

while (true) {
    if (add_task()) {
        create_tree();
        while (!all_solve) {
            if (!all_sent && free)
                send();
            if (recv())
                result_build();
        }
        build_timetable();
        recv_timetable();
    }
}

```

Fig. 3. Navigable canal management information system operation main flow pseudo-code.


```
while (true) {
    if (send()) {
        status = busy;
        while(!solve) {}
        recv();
        status = free;
    }
}
```

Fig. 4. Navigable canal management information system operation parallel flows pseudo-code.

3 Results

In presented navigable canal management information system operation algorithm was performed, all events, including the decisions taken, were recorded and, based on them, for a better control effect, a ship wiring schedule was formed.

However, with an increase in the options number, the computation time has increased, which suggests that the described algorithm can only be used for local optimization on current computer tools. In addition, it must be borne in mind that optimization affects only channels part and its forecast schedule depth is limited.

As a result of the navigable canal management information system operation algorithm parallelization within the acceptable time limits, the channel covered territory size and the planning period were increased. At the same time, the forecast schedule quality was improved, which had a positive effect on the navigable canal management information system operation performance.

In moving from a sequential algorithm to a parallel one, the volume choosing problem arises and subtask sent format to a solution at one of the parallel threads. The initial task volume and multithreaded system technical characteristics used determines the subtasks optimal size.

The variant tree is generated until the end nodes reaches number a predetermined value is proposed, which is calculated based on performance estimates set for all devices in the computing environment. This allows you to optimally distribute the load between any structure multi-threaded system all the resources.

The introduced atomic statements corresponding to program calculations allowed main events us to describe the parallel operation algorithm.

In Fig. 5 shows the navigable canal management information system operation stream diagrams: the only one for the main and the many for parallel flows. The nodes show the model processes basic states values, separated by commas.

Since the algorithm provides work with many parallel flows, consider a diagram demonstrating two parallel flows interaction dynamics of navigable canal management information system operation and one main one (Fig. 6).

The main and parallel flows interaction carried out at the sending timestamps the subtree (the navigable canal management information system operation main flow status block – «1.5, false») and receiving (the navigable canal management information system operation main flow block - «1.6, false») optimal control for n-th, must be synchronized. In the diagram, the flows synchronization is shown in bold lines.

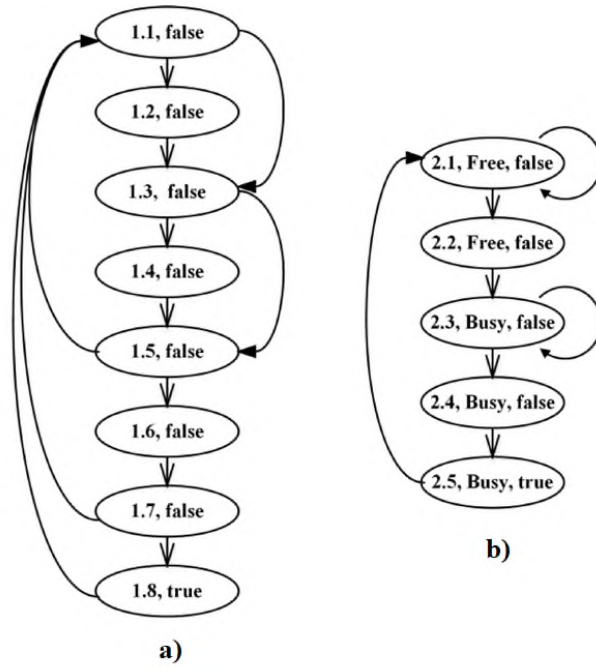


Fig. 5. The navigable canal management information system operation flow diagram: a) the main; b) parallel flows.

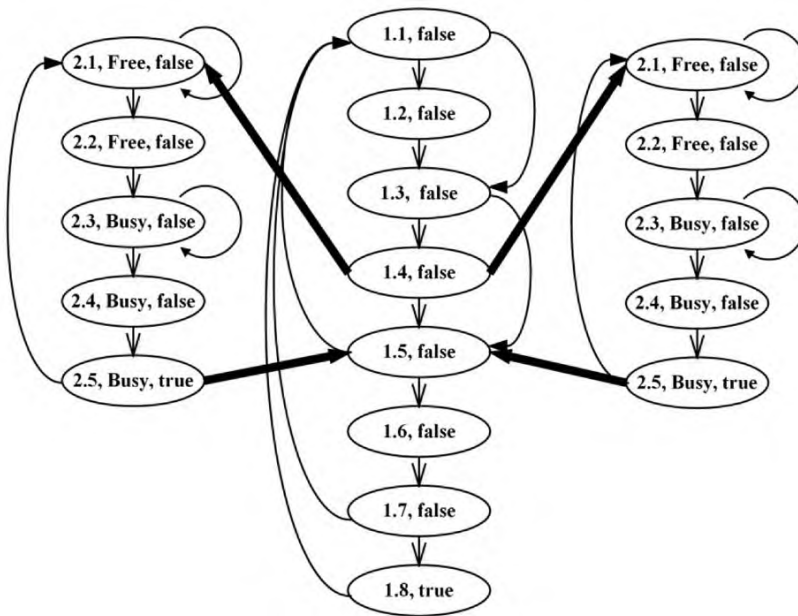


Fig. 6. Navigable canal management information system operation main interaction scheme with two parallel processing flows.

4 Discussion

Optimization the navigable channel operational management can be carried out using integer programming, however the dimensionality of such a problem does not allow it to be solved by exact methods using existing computer equipment. In this case, it is proposed to use the quasi-optimizing method based on the branch and bound method.

To reduce the options number that are searched, the method of «cutting off» non-promising solutions that lead to an unacceptable situation in the system or do not provide a given level of stability is used. It should be noted that the «clipping» of unacceptable situations is provided by a simulation locked navigable canal model.

With an increase in options number, the computation time increases, this affects the algorithm described can be used only for local optimization on current computer equipment. It is worth considering that optimization affects only channels part and its forecast schedule depth is limited.

Considering the fact that execution parallel algorithm time cannot be less than the execution its longest fragment time to ensure maximum efficiency when parallelizing the optimizing the ship's pass problem, the evaluating the devices performance must be chosen in such a way that, firstly, the calculation time for each stream is the same, and secondly, the generation subtasks time ratio and the search for optimal control was optimal. At the same time, it is also necessary to take into account the memory costs for storing the generated options tree.

After comparing algorithms effectiveness, can conclude that for subtasks formation, in parallel processing, it is necessary to use the information management system algorithm for the shipping channel in the horizontal passage nodes mode, and to search for the optimum in subtasks each on a separate thread more preferred is the vertical operation mode on the information management system.

5 Conclusions

When passage automated dispatching at the navigable shipping channel to increase productivity based on the universal parallelization software shell, it is necessary to use distributed data processing technologies, combining computers into a single computing cluster. This allows you to implement a hardware-software any computing power complex, the required determined level by the shipping channel parameters and if necessary, also increase it during the system operation

The single computing cluster construction makes it possible to increase the fault system tolerance, which is an important factor in the functioning of such an objects kind.

References

1. C.-S. Yang. Transportation Research Part E: Logistics and Transportation Review, **131**, 108–117 (2019). DOI: 10.1016/j.tre.2019.09.020.
2. K.F. Yuen, K.X. Li, G. Xu, X. Wang, Y.D. Wong. Transportation Research Part E: Logistics and Transportation Review, **128**, 316–332 (2019). DOI: 10.1016/j.tre.2019.06.014.
3. T. Tran, K. F. Yuen, K.X. Li, G. Balci, F. Ma, *Journal of Cleaner Production*, **243**, 301–316 (2020). DOI: 10.1016/j.jclepro.2019.118401.
4. Z. Łosiewicz, P. Nikończuk, D. Pielka. *Procedia Computer Science*, **159**, 2197–2205 (2019). DOI: 10.1016/j.procs.2019.09.394.

5. L. Liu, K.Y. Wang, T.L. Yip *Journal of Transport Geography*, **28**, 30–38 (2013). DOI: 10.1016/j.jtrangeo.2012.10.002.
6. L.B. Real, M. O'Kelly, G. Miranda, R.S. Camargo *Journal of Air Transport Management*, **73**, 95–112 (2018). DOI: 10.1016/j.jairtraman.2018.08.006.
7. T. Vanoutrive. *Case Studies on Transport Policy*, **7**, 489–496 (2019). DOI: 10.1016/j.cstp.2018.12.011.
8. Q. Zhang, S.Zheng, H. Geerlings, A. Makhloufi. *Transport Policy*, **77**, 46–57 (2019). DOI: 10.1016/j.tranpol.2019.03.001.
9. E. Arnold, H. Linke, R. Franke. *IFAC Proceedings*, **31**, 781–786 (1998). DOI: 10.1016/S1474-6670(17)41892-1.
10. E. Duviella, A. Doniec, H. Nouasse *Journal of Water Resources Planning and Management*, **144**, 04018059 (2018). DOI: 10.1061/(ASCE)WR.1943-5452.0000970.
11. P. Segovia, L. Rajaoarisoa, F. Nejjari, E. Duviella, V. Puig *Journal of Process Control*, **76**, 1–14 (2019). DOI: 10.1016/j.jprocont.2018.12.017.
12. R. Evans, L. Li, I. Mareels, N. Okello, M. Pham, W. Qiu, S.K. Saleem. *IFAC Proceedings Volumes*, **44**, 11459–11464 (2011). DOI: 10.3182/20110828-6-IT-1002.00451.
13. E. Duviella, B. Hadid. *IFAC-PapersOnLine*, **52**, 23–28 (2019). DOI: 10.1016/j.ifacol.2019.11.004.
14. K. Horvath, L. Rajaoarisoa, E. Duviella, J. Blesa, M. Petreczky, K. Chuquet. Springer International Publishing, Cham, 211–234 (2015). DOI: 10.1007/978-3-319-16133-412.
15. A.N. Alpatov. *Kibernetika I programirovanie*, **1**, 1–10 (2017). DOI: 10.7256/2306-4196.2017.1.2202.
16. J. Kim, J. Park, S. Hyun, D.H. Fleisher, K. SooKim. *Computers and Electronics in Agriculture*, **169**, 395–409 (2020). DOI: 10.1016/j.future.2019.11.040.
17. R.A. Cooke, S.A. Fahmy. *Future Generation Computer Systems*, **105**, (2020). DOI: 10.1016/j.compag.2019.105196.
18. L.F. Bittencourt, A. Goldman, E. R.M. Madeira, Nelson L.S. da Fonseca, R. Sakellariou. *Computer Science Review*, **30**, 31–54 (2018). DOI: 10.1016/j.cosrev.2018.08.002.
19. Y. Cao, P. Li, Y. Zhang. *Future Generation Computer Systems*, **88**, 279–283 (2018). DOI: 10.1016/j.future.2018.05.038.
20. A. Marowka *Journal of Parallel and Distributed Computing*, **138**, 199–210 (2020). DOI: 10.1016/j.jpdc.2019.12.020.
21. D. Hu, S. Yao, C. Duan, S. Li. *Journal of Hydrology*, **582**, 124531 (2020). DOI: 10.1016/j.jhydrol.2019.124531
22. C. Scheifele, A. Verl and O. Riedel. *Procedia CIRP*, **79**, 397–402 (2019). DOI: 10.1016/j.procir.2019.02.104.