DOI: https://doi.org/10.36910/6775-2524-0560-2024-55-40
UDC 519.8:004.42
Dymova Hanna, Candidate of Technical Sciences, Phd., Associate Professor
https://orcid.org/0000-0002-5294-1756
Oksana Larchenko, Candidate of Agricultural Sciences, Associate Professor
https://orcid.org/0000-0001-7857-0802
Kherson State Agrarian and Economic University, Kherson, Ukraine

USING THE MAX FLOW PROBLEM FOR BUSINESS PROCESSES

Dymova H., Larchenko O. Using the Max Flow Problem for Business Processes. The article is devoted to the use of optimization methods for managing business processes. Problems of synthesizing the structure and choosing parameters of systems, as well as problems of optimal control of them, are most often solved by reducing them to problems of finding a number of parameters so that the extremum of the quality function is achieved under a set of restrictions. A separate group of problems related to the choice of topology can be solved using the apparatus of graph theory. Problems of choosing channel capacity can be reduced to classical optimization problems, such as linear programming, transport problem, nonlinear programming, etc. The minimum flow problem is a special case of the transport problem, which is related to linear programming problems.

The paper analyzes the use of the maximum flow problem in economic systems, namely for planning business processes. The use of tasks to optimize business processes in various industries, such as logistics, production and finance, and personnel management, is described. A review of the developments and research of the maximum flow problem is carried out, a number of algorithms for solving this problem and their main points are given. The problems, the mathematical formulation of the problem, the algorithm for placing marks and the calculation of arc flows for each iteration are described in detail. An example of the flagging algorithm shows how maximum flow models can be used to analyze and optimize production processes. Maximum flow problems can identify bottlenecks in a production chain, optimize resource utilization, and plan production. Using the maximum flow problem when planning business processes will improve efficiency and increase production, help reduce costs and improve profitability, and ensure that products are in stock to meet demand.
Keywords: business processes, maximum flow, Ford-Fulkerson algorithm, source, stock, augmentative flow path.

Димова Г.О., Ларченко О.В. Використання задачі про максимальний потік для бізнес-процесів. Стаття присвячена використанню методів оптимізації для управління бізнес-процесами. Задачі синтезу структури та вибору параметрів систем, а також задачі оптимального управління ними найчастіше розв'язують, зводячи до задач знаходження низки параметрів так, щоб досягався екстремум функції якості при наборі обмежень. Окрему групу задач, які зв'язані з вибором топології, можна розв'язати із застосуванням апарату теорії графів. Задачі вибору пропускної здатності каналів можуть бути зведені до класичних задач оптимізації, таких як лінійного програмування, транспортної задачі, нелінійного програмування тощо. Задача про мінімальний потік являється окремим випадком транспортної задачі, що відноситься до задач лінійного програмування.

В роботі проаналізовано використання задачі про максимальний потік в економічних системах, а саме для планування бізнес-процесів. Описано використання задачі для оптимізації бізнес-процесів в різних галузях, таких як логістика, виробнича та фінансова сфери, управління персоналом. Зроблено огляд розробок та досліджень задачі про максимальний потік, наведена низка алгоритмів розв'язання цієї задачі та основні їх моменти. Детально наведено опис проблеми, математична постановка задачі, алгоритм розставлення поміток та розрахунок дугових потоків для кожної ітерації. Приклад роботи алгоритму розставлення поміток демонструє, як моделі максимального потоку можуть бути використані для аналізу та оптимізації виробничих процесів. Задачі про максимальний потік можуть визначити вузькі місця у виробничому ланцюгу, оптимізувати використання ресурсів та планувати виробництво. Використання задачі про максимальний потік при плануванні бізнес-процесів дозволить покращити ефективність та збільшення виробництва, допоможе зменшити витрати та покращити прибутковість, забезпечити продукцією на складі для задоволення попиту.
Ключові слова: бізнес-процеси, максимальний потік, алгоритм Форда-Фалкерсона, джерело, стік, аугментальний шлях потоку.

Formulation of the problem. Business process management is aimed at providing quality service to consumers (clients) [1]. For companies with a high degree of business diversification and a variety of partnerships, business process optimization provides solutions to the following tasks:

– determining the optimal sequence of functions performed, which leads to a reduction in the cycle time for the production and sale of goods and services, customer service, which results in an increase in capital turnover and an increase in all economic indicators of the company;

– optimization of the use of resources in various business processes, as a result of which production and distribution costs are minimized and an optimal combination of various types of activities is ensured;

– building adaptive business processes aimed at quickly adapting to changes in the needs of end consumers of products, production technologies, the behavior of competitors in the market and, consequently, improving the quality of customer service in a dynamic external environment;

– determination of rational schemes for interaction with partners and clients, and as a result, profit growth, optimization of financial flows.

The maximum flow problem is a special case of the transport problem and can be used to optimize business processes in various industries [2]. In logistics, maximum flow can be used to plan optimal delivery routes for goods or allocate resources between warehouses. This will help reduce delivery times and optimize the use of vehicles. In manufacturing, the maximum flow problem is used to optimize the flow of materials through production lines, thereby reducing congestion, increasing productivity, and reducing production costs. In the financial sector, the maximum flow problem helps banks efficiently distribute payment flows and reduce transaction processing time, that is, it is used to optimize the processing of financial transactions. In human resource management, a task can be used to optimize workflow flow and distribute tasks among work groups, allowing for more efficient use of work time and reducing task completion time. Consequently, the maximum flow problem is one of several deterministic flow models that are used to formulate and solve important economic problems.

Research analysis. The maximum flow problem has a rich history of research [2, 3]. The first research in this area was carried out by Lester Ford and Delbert Fulkerson (1955), who developed the Ford-Fulkerson algorithm, which is one of the most famous algorithms for solving the maximum flow problem. Later in 1972, Edmonds and Karp developed the Edmonds-Karp algorithm, which is more efficient than the Ford-Fulkerson algorithm for some types of graphs. There has also been research and development to solve this problem using different methods and algorithms. In 1988, one of the most effective algorithms for solving the maximum flow problem appeared - the push-relabel algorithm, developed by Alexander Rao and Andrew Goldberg. In 1995, Ulrich Feige and Hans-Wolfram Geertz proposed the potential method, which has become a powerful tool for analyzing and solving maximum flow problems. Kevin Busch developed a capacity-constrained flow algorithm in 2000 that can be used to solve maximum flow problems with additional constraints.

The study of the maximum flow problem is still ongoing. New algorithms and heuristics are constantly being developed, and existing methods are being improved. Let us list other researchers who have made significant contributions to this area - S.T. McCarthy, R.T. Rao, D.B. Kuhn, M.V. Frederiksen and S. Ramamurthy. The study of the maximum flow problem has had a significant impact on many fields, including computer science, operations research, mathematics, and engineering.

Presentation of the main material and justification of the obtained results. In the context of business process planning, consider a manufacturing enterprise that produces products at several stages: procurement of raw materials, production, assembly, packaging and shipment. Each stage has its own throughput - the maximum amount of products that can be processed in a certain period of time. The company's goal is to maximize the volume of products that can be shipped to customers.

There are many algorithms for solving the maximum flow problem, such as [1-4]:

- linear programming method: the problem is considered as a linear programming problem, where the variables are the flows along the edges, and the constraints are the conservation of the flow and the capacity limitation;
- Ford-Fulkerson algorithm: here you need to find any increasing path, increase the flow along all its edges by the minimum of their remaining capacities and repeat as long as there is an increasing path;
- Edmonds-Karp algorithm: the Ford-Fulkerson algorithm is executed, each time choosing the shortest increasing path (found by breadth-first search);
- Dinitz algorithm: is an improvement of the Edmonds-Karp algorithm. At each iteration, using breadth-first search, the distances from the source to all vertices in the residual network are determined. A graph is constructed containing only those edges of the residual network at which this distance increases by 1. Next, all dead-end vertices with edges incident to them are excluded from the graph until all vertices become non-dead-end; here a dead-end vertex is a vertex that does not enter or exit from any edge except the source and sink. The shortest increasing path is found on the resulting graph (this will be any path from s to t). Next, the edge with the minimum capacity is excluded from the residual network, dead-end vertices are again excluded, and so on while increasing paths still exist;
- preflow promotion algorithm: the algorithm operates on the preflow instead of a stream. The difference is that for any vertex, except for the source and sink, the sum of the flows entering it must be strictly zero (flow conservation condition), and for a preflow it must be integral. This amount is called excess flow into a vertex, and a vertex with positive excess flow is called overflowing. In

addition, for each vertex the algorithm stores an additional characteristic, height, which is an integral integer. The algorithm uses two operations: advancement and lifting. Promotion is possible when the edge belongs to the residual network, when it leads from a higher vertex to a lower one and the original vertex is full. Ascent is possible when the rising vertex is overcrowded, but none of the vertices to which the edges of the residual network lead from it are lower than it;

– "lift to top" algorithm: this is a variant of the previous algorithm, which in a special way determines the order of the promotion and lifting operations;
– binary blocking thread algorithm.

In this case, we will use the Ford-Fulkerson algorithm.

Let $\mathbf{G} = (\mathbf{N}, \mathbf{A})$ – be a directed network with one source $s \in \mathbf{N}$ and one sink $t \in \mathbf{N}$, and let the arcs $(i, j) \in \mathbf{A}$ have limited capacity. The maximum flow problem is to find flows along arcs belonging to the set $\mathbf{A}$, such that the resulting flow flowing from source s to sink t is maximum. It is assumed that an unlimited flow can enter the source and that for each intermediate node in the network the flow conservation condition is satisfied. This problem is nontrivial when the capacity U_{ij} of each arc represents a finite upper bound on the flow f_{ij} along that arc.

The maximum flow problem can be formulated as a linear programming problem, so the usual simplex method can be used to solve it. Let's consider another, more effective procedure for finding a solution to this problem. The algorithm starts with some feasible solution. The flagging procedure developed by Ford and Fulkerson [2, 5] is then performed to determine another allowable flow of larger magnitude. In this algorithm, nodes are considered as intermediate points of flow transmission, and arcs are considered as distribution channels. To formally describe the algorithm, it is necessary to introduce two basic concepts – markings and augmenting flow paths.

The node label is used to indicate both the flow value and the source of the flow causing a change in the current flow value along the arc connecting this source to the node in question [6]. If q_i units of flow are sent from node i to node j and cause an increase in flow along this arc, then we will say that node j is marked from node i with the symbol $+q_j$. In this case, node j is assigned the mark $[+q_j, i]$. Similarly, if sending q_j flow units causes flow to decrease along the arc, then node j is marked from node i with the symbol $-q_j$. In this case, node j is assigned the mark $[-q_j, i]$.

The current flow from node i to node j increases when q_j units of additional flow are sent to node j along an oriented arc (i, j) in the direction coinciding with its orientation. In this case, the arc (i, j) is called a direct flow [6]. A corresponding example is shown in Fig. 1.

The current flow from node i to node j is reduced when q_j flow units are sent to node j along an oriented arc (j, i) in the direction coinciding with its orientation. In this case, arc (j, i) is called reverse flow [6]. A corresponding example is shown in Fig. 2.

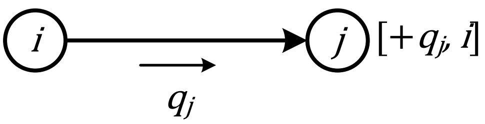


Figure 1 – Direct flow

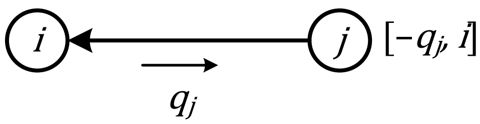


Figure 2 – Reverse flow

If node j is marked from node i and arc (i, j) is straight, then the flow along this arc increases and the value corresponding to the remaining unused capacity of the arc must be adjusted as necessary. This value is usually called the residual arc capacity. If a node is marked and a forward branch is used, then it can only have positive "residual capacity". In addition, node j can be labeled from node i only after the node has been assigned a mark.

An augmental flow path from s to t is defined as a connected sequence of forward and backward arcs along which several units of flow can be sent from s to t. The flow along each forward arc increases without exceeding its capacity, and the flow along each reverse arc decreases, while remaining negative. The augmented flow path is used to select a method of changing the flow in which the flow at node t increases and at the same time the flow conservation condition is not violated for each internal node of the network.

The maximum flow problem is often encountered in practice, and the number of nodes and arcs in the network often reaches several thousand. Therefore, to solve such problems it is necessary to use an efficient calculation procedure. Due to the simplicity of the formulation of the maximum flow problem, an effective recurrent algorithm for finding the optimal solution (maximum flow) was developed using the labeling procedure. Let us present an algorithm for placing marks for the maximum flow problem.

Let (i, j) be an oriented arc leading from node i to node j. Let us note in what cases the flow along a given arc can be increased [2, 6]. As noted above, the flow can be increased by q_j units if the arc (i, j) is a forward flow and node j is labeled $[+q_j, i]$. It remains to be seen when this occurs. Let us assume that the arc (i, j) is already assigned a flow $f_{ij} \geq 0$ ($f_{ij} \leq U_{ij}$). Obviously, the value q_j cannot exceed the residual capacity $U_{ij} - f_{ij}$. This is not enough to mark node j, since from node i it is not always possible to obtain $U_{ij} - f_{ij}$ flow units. Note that you can send as many flow units to node j as are added to node i, that is, at most q_i. Therefore, the flow along a straight arc (i, j) can be increased by the amount q_j, where $q_j = min[q_i, U_{ij} - f_{ij}]$.

In the same way, you can mark node j if the arc (j, i) is a reverse flow. Here the following question arises: is it possible to reduce the flow along the arc (j, i)? Obviously, this is only possible if $f_{ji} > 0$. This flow can be reduced at most by the number of flow units that can be taken from node i, that is, by the value q_i. Consequently, the flow along the reverse arc (j, i) can be reduced by the amount q_j, where $q_j = min[q_i, f_{ji}]$.

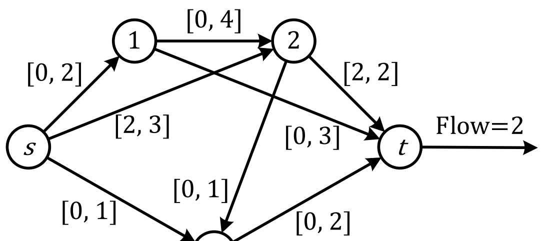
The marking algorithm works as follows. First, the source is assigned the mark $[\infty, -]$, indicating that a flow of infinitely large magnitude can flow from this node. Next, we look for the augmented flow path from the source to the sink, passing through the marked nodes. All nodes other than the source are initially unlabeled. Trying to reach the sink, we walk along forward and backward arcs and sequentially mark the nodes belonging to them [7]. The following two cases are possible:

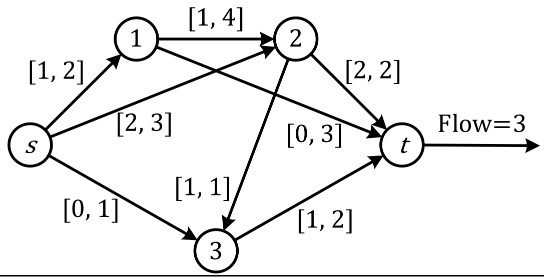
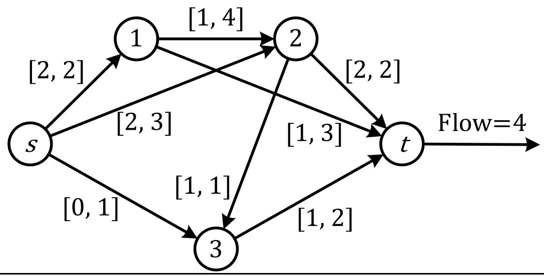
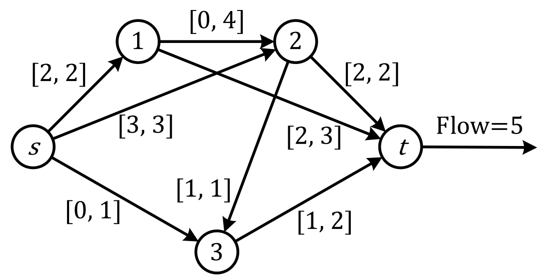
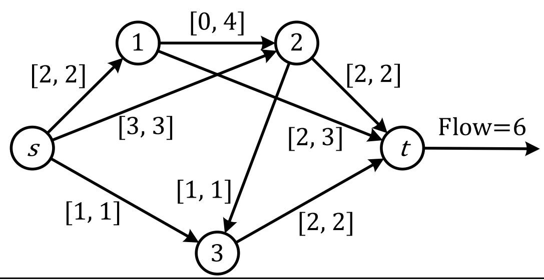
1. The stock t is labeled $[+q_t, k]$. In this case, the augmented flow path is found and the flow along each arc of this path can be increased or decreased by the amount q_t. After changing the arc flows, the current marks are erased and the entire procedure described above is performed again.

2. Stock t cannot be labeled. This means that the augment flow path cannot be found. Consequently, the constructed arc flows form an optimal solution (maximum flow).

To illustrate the operation of the labeling algorithm for the network shown in Table 1, the maximum flow that can flow from node s to node t will be found. Each arc (i, j) is assigned a mark $[f_{ij}, U_{ij}]$, where f_{ij} is the current value of the arc flow, and U_{ij} is the arc capacity. Initially, the magnitudes of all arc fluxes are assumed to be zero. When performing each iteration, you need to mark the stock t. This problem is solved in 6 iterations, the results of each of which are shown in Table 1.

Table 1 – Operation of the marking algorithm

Iteration number	Steps	Description of the procedure	Graph
1	1	Assign to node s the mark $[\infty, -]$	
	2	Assign to node 2 the mark $[+3, s]$	
	3	Assign to node t the mark $[+2,2]$	
	4	Changing arc flows: $f_{s2} = 2, f_{2t} = 2$	
2	5	Assign to node s the mark $[\infty, -]$	
	6	Assign to node 1 the mark $[+2, s]$	
	7	Assign to node 2 the mark $[+2,1]$	

<table>
<tr><td rowspan="3"></td><td>8</td><td>Assign to node 3 the mark $[+1,2]$</td><td rowspan="3">
</td></tr>
<tr><td>9</td><td>Assign to node t the mark $[+1,3]$</td></tr>
<tr><td>10</td><td>Changing arc flows: $f_{s1} = 1, f_{12} = 1, f_{23} = 1, f_{3t} = 1$</td></tr>
<tr><td rowspan="4">3</td><td>11</td><td>Assign to node s the mark $[\infty, -]$</td><td rowspan="4">
</td></tr>
<tr><td>12</td><td>Assign to node 1 the mark $[+1, s]$</td></tr>
<tr><td>13</td><td>Assign to node t the mark $[+1,1]$</td></tr>
<tr><td>14</td><td>Changing arc flows: $f_{s1} = 2, f_{1t} = 1$</td></tr>
<tr><td rowspan="5">4</td><td>15</td><td>Assign to node s the mark $[\infty, -]$</td><td rowspan="5">
</td></tr>
<tr><td>16</td><td>Assign to node 2 the mark $[+1, s]$</td></tr>
<tr><td>17</td><td>Assign to node 1 the mark $[-1,2]$</td></tr>
<tr><td>18</td><td>Assign to node t the mark $[+1,1]$</td></tr>
<tr><td>19</td><td>Changing arc flows: $f_{s2} = 2, f_{12} = 0, f_{1t} = 2$</td></tr>
<tr><td>Iteration number</td><td>Steps</td><td>Description of the procedure</td><td>Graph</td></tr>
<tr><td rowspan="4">5</td><td>20</td><td>Assign to node s the mark $[\infty, -]$</td><td rowspan="4">
</td></tr>
<tr><td>21</td><td>Assign to node 3 the mark $[+1, s]$</td></tr>
<tr><td>22</td><td>Assign to node t the mark $[+1,3]$</td></tr>
<tr><td>23</td><td>Changing arc flows: $f_{s3} = 1, f_{3t} = 2$</td></tr>
<tr><td rowspan="2">6</td><td>24</td><td>Assign to node s the mark $[\infty, -]$</td><td rowspan="2"></td></tr>
<tr><td>25</td><td>None of the nodes can be marked, so the maximum flow is 6</td></tr>
</table>

Conclusions and prospects for further research. The Ford-Fulkerson algorithm is one of the most famous algorithms for solving maximum flow problems. The paper presents a general description of the problem, a mathematical formulation and a description of the algorithm for solving the problem. Using maximum flow models can help businesses improve their efficiency and profitability, calculate flow patterns to reduce costs and provide better customer service.

References

1. Saaty T.L. & Kearns K.P. (1985) Analytical Planning. The Organization of Systems. Pergamon Press. Oxford, New York, Toronto, Sydney, Frankfurt.
2. Phillips D.T. & Garcia-Diaz A. (1981) Fundamentals of Network Analysis. Prentice-Hall, Inc., Englewood Cliffs, N.J.
3. Anderson J.A. (2001) Discrete Mathematics with Combinatorics. Prentice Hall, Upper Saddle River, New Jersey.
4. Murtagh B.A. (1981) Advanced Linear Programming: Computation and Practice. McGraw-Hill International Book Company, New York, London, Madrid.

5. Dymova H. (2024) Development of a Software Application Algorithm for Solving Computer Network Optimization Problems. Débats scientifiques et orientations prospectives du développement scientifique: c avec des matériaux de la VI conférence scientifique et pratique internationale, Paris, 1er Mars 2024. Paris-Vinnytsia: La Fedeltà & UKRLOGOS Group LLC. DOI: https://doi.org/10.36074/logos-01.03.2024.051.
6. Dymova H. & Larchenko O. (2020) Rozrobka komp″yuternoyi prohramy rozv″yazannya zadach merezhevoyi optymizatsiyi [Development of a Computer Program for Solving Network Optimization Problems]. Computer-Integrated Technologies: Education, Science, Production. Lutsk. No. 41. DOI: https://doi.org/10.36910/6775-2524-0560-2020-41-23. [in Ukrainian].
7. Dymova H. (2023) Application of Characterization Analysis Methods to Investigation of Logical Networks Structures. Theoretical and Empirical Scientific Research: Concept and Trends with Proceedings of the V International Scientific and Practical Conference. Oxford, United Kingdom: European Scientific Platform. DOI: https://doi.org/10.36074/logos-23.06.2023.34.

Список бібліографічного опису

1. Saaty T.L. & Kearns K.P. (1985) Analytical Planning. The Organization of Systems. Pergamon Press. Oxford, New York, Toronto, Sydney, Frankfurt.
2. Phillips D.T., Garcia-Diaz A. (1981) Fundamentals of Network Analysis. Prentice-Hall, Inc., Englewood Cliffs, N.J.
3. Anderson J.A. (2001) Discrete Mathematics with Combinatorics. Prentice Hall, Upper Saddle River, New Jersey.
4. Murtagh B.A. (1981) Advanced Linear Programming: Computation and Practice. McGraw-Hill International Book Company, New York, London, Madrid.
5. Dymova H. (2024) Development of a Software Application Algorithm for Solving Computer Network Optimization Problems. Débats scientifiques et orientations prospectives du développement scientifique: c avec des matériaux de la VI conférence scientifique et pratique internationale, Paris, 1er Mars 2024. Paris-Vinnytsia: La Fedeltà & UKRLOGOS Group LLC. DOI: https://doi.org/10.36074/logos-01.03.2024.051.
6. Димова Г., Ларченко О. (2020) Розробка комп’ютерної програми розв’язання задач мережевої оптимізації. Науковий журнал "Комп’ютерно-інтегровані технології: освіта, наука, виробництво", (41). DOI: https://doi.org/10.36910/6775-2524-0560-2020-41-23.
7. Dymova H. (2023) Application of Characterization Analysis Methods to Investigation of Logical Networks Structures. Theoretical and Empirical Scientific Research: Concept and Trends with Proceedings of the V International Scientific and Practical Conference. Oxford, United Kingdom: European Scientific Platform. DOI: https://doi.org/10.36074/logos-23.06.2023.34.