Solution of task on the minimum cost data flow based on bionic algorithm

A Yu Poluyan, O A Purchina, D D Fugarov, A A Golovanov, O V Smirnova

Don State Technical University, Rostov-on-Don, Russian Federation

Abstract: The paper presents the solution of task on the minimum cost data flow in static and dynamic setting based on evolutionary algorithms. The distinctive feature of the algorithm is the use of joint evolution, choice of evolution models (use of micro-, macro- and metaevolutions), adjustment to external environment, hierarchical management of genetic and evolutionary search, local search of solutions and application of all modified genetic operators based on evolutionary strategy and search methods. The paper illustrates the example of task on the recommended data flow and a method of adjusting the data transfer process to recommended parameters.

1. Introduction

The tasks on the minimum cost flows hold a special place among priority areas of study in the field of information technologies since the solution of such tasks is currently relevant for the industry, transport and other spheres. This task refers to the satisfaction of needs with known requirements and means of needs satisfaction. Any enterprise is interested in minimizing its costs. For example, the organization arranges the delivery of food provided it knows the volume, time, cost of transportation, cost restrictions, and it is interested to save its costs. Based on such data it is possible to build a task on the minimum cost taking into account all data and imposed restrictions.

The task can be formulated in terms of the graph theory as the task on the minimum cost flow. Data on the adjacency of graph nodes and cost of data transmission are constant in the creation of the static model of a task on the minimum cost flow. Two graph nodes i and j are connected by an edge (i, j) only if it is possible to transfer data from the node A_i to the node A_j. Each such edge is assigned a weight corresponding to transmission time of the information unit from A_i to A_j. Thus, the adjacency matrix \|a_{ij}\|_{n,m} describes possible directions of data transmission, while the matrix \|c_{ij}\|_{n,m} – the maximum permissible number of data transferred from A_i to A_j. The matrix \|v_{ij}\|_{n,m} reflects the number of data actually transferred to the unit of time [1].

The graph described by the matrix \|a_{ij}\|_{n,m} is directed and can be cyclic. Since while solving optimization tasks using the obtained graph it is much more convenient to use the acyclic model, the initial cyclic graph may be presented in the acyclic equivalent form. Graphs are meant to be equivalent in terms of unambiguity of algorithms interpreted by them thus saving the original information [2]. The task of receiving the acyclic graph can be set in terms of integer linear programming – as a covering problem. The received acyclic graph may be made consecutive, i.e. a graph satisfying three properties [3]:

• the graph has an ordinal function P(x) with values 0, 1..., n, where n – quantity of graph layers;
• the graph nodes being in one layer are not incidents to each other;
the number of edges from a layer $S_i$, $i=1, ..., n-1$ coincides with the number of edges of a layer $S_{i+1}$, $i=1, ..., n$ while only edges from $S_i$ layers with numbers $k<i$, $k=1, 2, ..., i-1$ can enter a layer $S_i$.

Let us modernize the matrix $[a_{ij}]_{m,n}$ having added as $(m+1)$ column a vector column, which coordinates are calculated according to the formula:

$$a_{i,m+1} = \sum_{j=1}^{n} a_{i,j},$$

where $i=1, 2, ..., n$.

Nodes $k$, for which $a_{k,m+1}=0$ have no outgoing edges and form the first layer $S_1$, for which $P(S_1)=1$ is fair.

The values of the column vector $a_{k,m+2}$ are defined by the formula:

$$a_{i,m+2} = a_{i,m+1} - \sum_{a_i \in S_k} a_{i,j},$$

where $i=1, 2, ..., n$.

Zero values of column $a_{i,m+2}$ constitute the second layer $S_2$. For the nodes of $S_2$ layer $P(S_2)=2$ is fair. The subsequent layers are calculated similarly, the total number of all layers does not exceed $n$ [4].

Fake nodes and edges are entered on the basis of matrix $[a_{ij}]_{m,n}$ and function $P$. If $a_{ij} = 1$, i.e. from $a_i$ node the edge leads to node $a_j$ and $P(a_j) < P(a_i)$, then the nodes connected by edges of zero weight are entered into all intermediate layers. The first fake edge going from $a_i$ to the first fake node acquire weight $c_{ij}$. If values of ordinal function $P$ correspond to nodes $i$ and $j$, then the fake nodes and edges are not entered [5].

In the received consecutive count, we split the bichromatic graph into two subgraphs, for which the problem of optimization is solved on the basis of the Ford’s method and which serves the basis for initial solutions (subpopulations $A_1, ..., A_{\text{Count}}$) [6]. It is possible to increase the operating speed of a bionic algorithm and improve the quality of obtained solutions through parallelization of the task solution.

**Statement of Purpose**

Fig. 1 shows the scheme of an algorithm to search the solution of a task on the minimum cost flow. This scheme uses the ideas of joint evolution, choice of evolution models (use micro-, macro-metaevolutions), adjustment to external environment, active interaction with external environment, hierarchical management of genetic and evolutionary search, local search of solutions and application of all modified genetic operators based on greedy strategy and search methods. On Fig. 1 BS1, BS_Count – bionic search for parts of populations, which develop in parallel and independently. At some stage of work there is an exchange of a part of individuals between parts of subpopulation on the basis of the modified operator of mutation. So, it can continue up to the completion of an algorithm [7].

Let us consider the bionic search (BS_1, ..., BS_Count) is more details:

1. Input of boundary conditions and optimality criterion; calculation of objective function for $A_1$, ..., $A_{\text{Count}}$.
2. Check of subpopulation on a hit condition into a local optimum; if the condition is satisfied, then go to item 6, otherwise follow actions in item 3.
3. Realization of genetic search (GS), work of all modified genetic operators.
4. Determination of GS stop criterion – number of iterations of the modified GA. In case the criterion is satisfied, the transition to the evolutionary search (ES) is performed.
5. Realization of the migration operator, formation of new population taking into account the optimal solutions received at GS stage.
6. Modeling of evolutionary search depending on input parameters (choice of the strategy of mutation operator). The competition between all individuals to be included into the following population is a powerful tool of adjustment to ES.
Figure 1. Scheme of an algorithm of task solution on the minimum cost flow.

7. Definition of ES stop criterion – number of iterations of the modified mutation operators (MO). In case the criterion is satisfied, the transition to GS modeling is performed, or the optimal solution is developed.

8. Analysis of solutions received while performing the GS. As a result of the carried-out analysis each solution (individual) is assigned a certain rank (perspective, unpromising, trivial, etc.). The ranging of the current population of alternative solutions carried out in the considered block allows increasing the efficiency of bionic search due to bigger structuredness of a variety of alternative solutions and gives the chance of dynamic regulation of the search direction.

9. Realization of the migration operator, formation of a new population taking into account optimal solutions received at the ES stage.

10. Assessment of the entire population.

11. Realization of the modified operator of reduction, formation of a new population taking into account optimal solutions.

12. Check on preliminary convergence, in case the condition is satisfied, the adaptation block is applied to the entire population, otherwise go to item 13.
13. Determination of a stop criterion of the bionic algorithm (BA) – number of iterations and/or operating time of an algorithm.

One of the key problems is to define the required optimum quantity of chromosomes from subpopulations during the work of the parallel bioinspired algorithm. Time and profitability of algorithm operation depends on the size of the population. It is suggested to use the following formula for the selection of the required number of chromosomes:

\[ n = \frac{t^2 \sigma^2}{\Delta^2} \]  

(3)

where  
- \( n \) – quantity of selected chromosomes for migration;  
- \( t \) – coefficient determined by the Table of Laplace Transforms, \( F(t)=p \), where \( p \) – set probability of the migration operator determined by a person making the decision;  
- \( \Delta \) – margin of sampling error, represents a limit above the absolute value \( |\varepsilon| < \Delta \);  
- \( \sigma \) – mean root square deviation.

The conducted tests show that the use of the modified migration operator allows reducing the operating time of an algorithm.

**Proposed Approach**

When solving a static problem of optimization, the adjustment machine can be used to manage search in case of unclear command “cost of flow transfer shall be close to \( S \)” (Fig. 2).

![Scheme of transition of the adjustment machine solving a static problem.](image)

**Figure 2.** Scheme of transition of the adjustment machine solving a static problem.

Depending on the proximity of the received intermediate solution to \( S \) the adjustment machine makes a decision to shift to a new status. The alternative \( A_1 \) means the stop of an algorithm, and \( A_2 \) – the need of local review of an intermediate solution, and \( A_3 \) – full review of an intermediate solution.

In case of the dynamic model the matrices take the following form: \( \|a_{ij}(t)\|_{n,m} \), \( \|c_{ij}(t)\|_{n,m} \) and \( \|x_{ij}(0)\|_{n,m} \) respectively. Let us consider the situation when the dynamism of a model means the change of transmitted flows or transfer cost of the set flows. In such situations the boundaries of permissible values of speed and cost of data transmission shall be set not in accurate but in indistinct form. For example, the admissible data transmission rate can be set on the basis of the known formula of indistinct proximity \( \mu_s(b) \) by variable \( b \) to the set value \( x \) [8, 9]:

\[ \mu_s(b) = e^{-\tau(x-b)^2} \]  

(4)

where \( \tau \) depends on the required uncertainty degree \( \mu_s(b) \) and is defined by the following expression:

\[ \tau = -\frac{4\ln 0.5}{\theta^2} \]  

(5)
where $\theta$ – distance between transition points, i.e. such $b_1$ and $b_2$ values, at which $\mu_x(b_1)=\mu_x(b_2)=0.5$.

Then inequality $\mu_x(b)\geq \mu_{\text{дон}}$ where $\mu_{\text{дон}}$ – minimum admissible degree of membership can be the criterion of compliance to the current data transmission rate. If inequality $\mu_x(b)\geq \mu_{\text{дон}}$ is not satisfied, then this indicates the need to review the routes of data transmission in order to reduce irregularity of decision-making, which may be caused by short-term peak loads. This makes it possible to use the adjustment machine. The adjustment machine supports three alternatives $A_1, A_2, A_3$ (Fig. 3). $S_{11}$ condition corresponds to $A_1$ alternative, $S_{21}$ and $S_{22}$ – to $A_2$ alternative, $S_{31}$ to $A_3$ alternative. Alternative $A_1$ means the invariance of available distributions of data transmission routes; $A_2$ includes the need for local reoptimization of document flows, and alternative $A_3$ – full reoptimization of document flows.

The transition between the conditions of the adjustment machine may happen on the basis of indistinct results of the data transmission analysis. The table below shows an example of rules of control signals.

| No. | Current alternative | Condition | Recommended alternative | Control signal |
|-----|--------------------|-----------|------------------------|---------------|
| 1   | A1                 | $\mu_x(b)\geq \mu_{\text{дон}}$ | A1           | +             |
| 2   | A1                 | $\mu_x(b)\leq \mu_{\text{дон}}$ | A2           | -             |
| 3   | A2                 | $\mu_x(b)\geq \mu_{\text{дон}}$ | A1           | +             |
| 4   | A2                 | $\mu_x(b)\leq \mu_{\text{дон}}$ | A3           | -             |
| 5   | A3                 | $\mu_x(b)\geq \mu_{\text{дон}}$ | A1           | +             |
| 6   | A3                 | $\mu_x(b)\leq \mu_{\text{дон}}$ | A3           | -             |

In compliance with the rules described in the table, the adjustment machine changes its state.

In case there is a need to perform data transmission optimization, the adjustment machine shifts into $A_2$ condition corresponding to alternative optimization with quicker convergent algorithm. If the algorithm converges beforehand, more resource-intensive modifications are used.

**Conclusions**

The study proposes the adaptive bionic algorithm to solve the task on the minimum cost data flow in static and dynamic setting. The distinctive feature of the algorithm is the use of adjustment machines to define the need and method of modification of intermediate solutions, as well as to make a decision on the change of the earlier received solution.
The efficient solution of such problems through population algorithms includes two key components:

- definition of the optimum configuration (architecture and parameters) of the algorithm;
- choice of the optimum computing structure, on which the algorithm of the configuration defined earlier, will be implemented.

Before start and in the course of the algorithm implementation its optimum configuration may be unknown a priori. Therefore, the proposed methods of adaptive management of the search of optimal solutions are quite relevant. If the necessary quantity of individuals in populations is known, then, from the point of view of the management of computing resources, the problem of the choice of a hardware component, on which the used algorithm will be implemented for the smallest time, seems efficient. The choice between the implementation of an algorithm on GPU or CPU can be an example of such task. Therefore, there is a need for adaptive definition of boundaries of more effective implementation of each version of an algorithm.

A large number of works is devoted to adjustment of architecture and parameters of optimizing algorithms [10, 11]. For example, the work [10] describes the adaptive genetic algorithm allowing changing the speed of mutation and crossing-over on the basis of the assessment of results of the corresponding posterity. The methods of adaptive management (change of architecture of evolutionary search, change of parameters of evolutionary algorithms, change of adjustment strategy) proposed within the project expand the adjustment range while solving the given task thus supplementing foreign studies.

The GPU technology implies parallelization of the solution search [11]. The methods of adaptive management of the process of search of optimal solutions will provide for more rational use of available CPU and GPU resources. This significantly improves the results received abroad.

References

[1] Chernyshev Yu O Poluyan A Yu 2009 Solution of the optimization task based on parallel bionic search SFU. Technical science 4 (93) 34-39
[2] Poluyan A Yu 2008 Adaptive genetic algorithm for the solution of the optimization task based on the strategy of elitism SFU. Technical science 9 (Taganrog: TTI SFU publishing house)
[3] Chernyshev Y O 2015. Swarm-intelligence-based algorithm of connections permutation between pins Journal of Theoretical and Applied Information Technology 81(3) 466-473
[4] Solomentsev K Yu Fugarov D D 2018 Interference elimination in digital controllers of automation systems of oil and gas complex. J. Phys. Conf. Ser 1015 032179.
[5] Poluyan A Y Fugarov D D 2018. Adaptive algorithm of selecting optimal variant of errors detection system for digital means of automation facility of oil and gas complex. J. Phys. Conf. Ser. 1015 022013.
[6] Fugarov D D Gerasimenko Y Y 2018. Methods for Revealing Hidden Failures of Automation System for Technological Processes in Oil and Gas Sector. J. Phys. Conf. Ser. 1118 012055.
[7] Chernyshev Yu O Ostroukh E N 1977 Solution of the optimization task on algorithms by network methods In Electronics and modeling vol. 15 pp. 34-36 (Kiev: Naukova dumka)
[8] Chernyshev Y O 2015. Swarm-intelligence-based algorithm of connections permutation between pins Journal of Theoretical and Applied Information Technology 80(1) 13-20
[9] Lipsky V 1988 Combinations theory for programmers (Moscow: Mir) p 200.
[10] Goldberg D E 1989 Sizing Populations for Serial and Parallel Genetic Algorithms the Third International Conference on Genetic Algorithms (San Mateo, CA: Morgan Kaufman) pp 70-79
[11] Mitchell R Frank E 2017 Accelerating the XGBoost algorithm using GPU computing Peer J Computer Science 3 127. DOI: 10.7717/peerj-cs.127.