











The known MFP-algorithms suppose predefined fixed arcs-weights, and thus, do not correctly work on the FWG-model with flexible scheduled arcs. We propose here an alternative approach to MFP study in the form of the direct and inverse Maxflow task.

The ordinal direct Maxflow task (O-DMT) we set as ‘how to construct the free-oriented ST-planar graph  $G(F)$  with minimal edges` capacities to provide the given total flow distribution  $F$  over the set of ST-paths. The inverse Maxflow task (IMT) we set as ‘how to find the maximal feasible ST-flow distribution  $F(G)$  over the set of ST-paths on the given free-oriented ST-planar graph  $G$ .

Generic inverse maxflow task IMT implies scanning all the ST-paths on the network graph topology. The total number of paths  $P(V)$  as function of vertices quantity  $V$  has a faster than polynomial growth even for planar graphs. Such computational tasks are referred to as nondeterministic polynomial complete problems (NPC) [13].

To solve complex combinatorial optimization problems of NPC-type, heuristic approaches are often used, which do not guarantee the best results, but may have less complexity, calculation time and produce adequate outcomes [18].

One of the famous and widely used heuristic method is Pontryagin maximum principle. This is choosing a set of necessary extreme conditions kept at any point of the object phase-track or the algorithm iteration, aimed to curtail the initial solutions diversity to a compact set of candidates for optimum. In some cases, the necessary conditions prove sufficient [9].

Solving complex NPC problem by heuristic method needs reliable proof the result obtained. The rigorous proof not always feasible, and empirical verification often necessary using various testing tools. Here, we propose an idea ‘Verify complex inverse task algorithm by simple direct task samples generation’.

The spoken above ordinal direct Maxflow task O-DMT is rather simple for particular given flow  $F$ -distribution over ST-graph  $G$ . In general, the direct Maxflow task (DMT) turns into non-trivial case of ‘generation a comprehensive sequence  $\{F, G\}$  with input  $F$  and output  $G$  for IMT-testing with inversed input/output  $\{G, F\}$ ’.

Today, more and more human tasks addressed to artificial intelligence (AI) with digital neural networks, that capable to be taught by machine learning (ML) techniques. The matter is, how to construct training samples sequences (TSS) for AI-models comprehensive teaching.

Let  $S = (G, F)$  a distinct DMT-task output, which can be used for an IMT-case testing. In these terms, we define  $\{S\} = \{(G, F)\}$  as the testing/training samples sequence (TSS) for testing the inverse Maxflow task algorithms on the free-oriented ST-planar graph.

Towards the Maxflow problem in the context of AI, we propose the DMT-output generator of IMT-tests  $\{G, F\}$  to be used as a training samples sequence (TSS) for machine learning of an AI-model. The core issue of the direct Maxflow task (DMT) is, how many samples  $\{S\}$  needed for complete testing the IMT-algorithms or exhaustive MFP machine learning.

## 5 DEFINITION OF THE DIRECT MAXFLOW TASK IN TENSOR FORM

We formulate a particular case of direct Maxflow task (DMT) on the bi-pole planar FWG-graph as following: ‘Find metric  $M$  of network graph  $G(V)$  with arbitrary number of vertices  $V$ , which is relevant to given flow distribution  $F(P)$  over the set  $P(V) = \{p\}$  of paths  $p \in P(V)$  in the given ST-planar free-oriented network graph  $G$  with complete topology  $T$ ’. The ‘complete topology’ means, that all the possible edges  $E_{PG}$  of the ST-planar graph  $G(V)$  are included (Figure 3).

Consider the complete planar graph  $G(V)$  with  $V=6$  vertices (Figure 1). The graph topology  $T$  is presented by the set of 12 edges  $E_{PG} = \{e\}$  identified by their digital values  $1 \div 12$  in matrix graph (Figure 2). All the edges  $e \in E_{PG}$  are mutually independent; let each of them be unitary (topological) vector  $e$ , and the set  $\mathbf{E} := \{e\}$  be the Euclid vector basis. Graph  $G(6)$  has 8 paths  $P(6) = 8$ , and each path  $p \in P(V)$  possess at least one backward-unique edge  $e_f$  to carry augment flow  $f(p)$ . The set of 8 shortest paths is a system of vectors (tensor)  $\mathbf{P} = \{p\}$  in matrix view (Figure 4); here, empty cells are zeros,  $p(1) = e(1)$ ;  $p(2) = e(2) + e(3)$  and so on. In this case,  $\mathbf{P}$  includes: a single 1-hop path, four 2-hop paths and three 3-hop paths.

The matrix of scalar products  $\{p(k) \times p(m)\}$  is *paths-metric tensor*  $M_P$  of  $\mathbf{P}$  (Figure 5). It is evident, that the first five vectors of  $\mathbf{P}$  are mutually orthogonal, as well as vectors number 6 and 8. The system  $\mathbf{P}$  with  $M_P > 0$  we call the *path’s tensor*. It is easy to show, that  $M_P$  is positive matrix [25].

Let  $\{f_p(k)\}$  be the distribution of the entire ST-flow  $F_\Sigma$  over the paths  $p(k)$  of the graph  $G(V)$ ,

Figure 4;  $F_{\Sigma} = \sum\{f_p(k)\}$ ;  $\mathbf{f} := \{\sqrt{f_p(k)}\}$  the *flow row-vector* with scalar product  $(\mathbf{f} \times \mathbf{f}) = F_{\Sigma}$ ;  $\mathbf{F} := \text{diag}(\mathbf{f})$  the diagonal matrix denoted as *flow-tensor*.

p(k)	e(n)												
	n	1	2	3	4	5	6	7	8	9	10	11	12
k													
1		1											
2			1	1									
3					1		1						
4								1		1			
5											1		1
6			1			1	1						
7							1	1	1				
8								1				1	1

Figure 4: System P of path' vectors in ST-planar graph.

The metric tensor  $M_F = (\mathbf{F} \times \mathbf{F})$  is distribution of the total flow  $F_{\Sigma}$  over the paths  $\mathbf{P} = \{\mathbf{p}\}$ ;  $F_{\Sigma} = \text{trace}(M_F) = (\mathbf{f} \times \mathbf{f})$ . On this premise, we define the *flow-paths tensor*  $\mathbf{F_P} := \mathbf{F} \times \mathbf{P}$ . For the graph in Figure 1, the  $\mathbf{F_P}$  is (8×12) matrix, where each k-row element is multiplied by the correspondent  $\sqrt{f_p(k)}$  value.

k	n							
	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	0
2	0	2	0	0	0	0	1	0
3	0	0	2	0	0	0	1	1
4	0	0	0	2	0	0	0	1
5	0	0	0	0	2	0	0	1
6	0	1	1	0	0	3	1	0
7	0	0	1	1	0	1	3	1
8	0	0	0	0	1	0	1	3

Figure 5: Metric tensor  $M_P$  of path's P vector system.

The convolution  $\mathbf{g} := \mathbf{f} \times \mathbf{F_P}$  we define as network *flow-load metric*. For the graph G in Figure 1,  $\mathbf{g}$  is the row-set with 12 flow-loads of G-edges. If  $f_p(k) \equiv 1$  and  $\mathbf{P}$  in Figure 4, then  $\mathbf{g} = (1, 2, 1, 1, 1, 3, 3, 1, 1, 1, 1, 2)$ .

Now, the direct Maxflow task for ST-planar free-oriented graph G with complete topology T lies in generating a relevant manifold of path's tensors  $\mathbf{P}$  on T, along with the metric tensors  $M_F$ . Here, the shortest-paths criteria or the maximal paths-diversity can be applied. This results in calculation the tensor  $\mathbf{F_P} = \mathbf{F} \times \mathbf{P} = \text{diag}(\mathbf{f}) \times \mathbf{P}$  and row-set  $\mathbf{g} = \mathbf{f} \times \mathbf{F_P}$ :

$$\text{DMT: } (\mathbf{P}, M_F)_T \rightarrow (\mathbf{g})_T.$$

The graph  $G(T, \mathbf{g})$  with topology T and metric  $\mathbf{g}$  is the input for the inverse Maxflow task, while the output is  $(M_F, \mathbf{P})$ . Thereby, a comprehensive set  $\{G(T, \mathbf{g}) / (M_F, \mathbf{P})\}$ , obtained by the routine direct

Maxflow task solution, can be used for testing the non-trivial algorithms of the hard inverse Maxflow task, as well as a training sample sequence (TSS) for machine learning of an AI-model:

$$\text{IMT: } (\mathbf{g})_T \rightarrow (\mathbf{P}, M_F)_T.$$

Ultimately, the direct Maxflow task (DMT) as a first part of the whole Maxflow problem (MFP) for data flows simulation in telecommunication networks can be given by the following formalism.

- 1) Bring the network topology to the normalized view of the ST-planar graph  $G(V)$  with V vertices (see Figures 1 and 2).
- 2) Generate the path's tensor  $\mathbf{P}(G)$  and flow-tensor  $\mathbf{F} = \text{diag}(\mathbf{f})$ ,  $\mathbf{f} = \sqrt{f_p(k)}$  (see Figure 4).
- 3) Calculate the flow metric-tensor  $M_F = (\mathbf{F} \times \mathbf{F})$ .
- 4) Get flow-paths tensor  $\mathbf{F_P} = \mathbf{F} \times \mathbf{P}(G)$ .
- 5) Count the graph  $G(V)$  metric  $\mathbf{g} = \mathbf{f} \times \mathbf{F_P}$ .
- 6) Fix  $(\mathbf{P}, M_F)_T \rightarrow (\mathbf{g})_T$  as a DMT result sample.
- 7) Use  $\{(\mathbf{g})_T \rightarrow (\mathbf{P}, M_F)_T\}$  as IMT/TSS sequence.

## 6 DISCRETE ANALYSIS OF THE INVERSE MAXFLOW TASK ON THE FREE-ORIENTED ST-PLANAR NETWORK GRAPH

Following the definitions in Section 5, here we introduce the inverse Maxflow task (IMT) on the free-oriented ST-planar weighted network graph  $G(V)$  with V vertices, complete topology T and given metric  $\mathbf{g}$ , as the objective: "Find the feasible total flow value  $F_{\Sigma}$  between the two open poles S, T, along with the path's tensor  $\mathbf{P}(G)$  and flow-path tensor  $\mathbf{F_P}$ ".

This type of discrete optimization tasks belongs to the known class of NPC combinatorial problems. An obvious but worst strategy for exact solving the formulated above IMT task is as follows.

Let  $P_i = \{p_k\}_i$  the set of ST-paths on the complete ST-planar graph G, where each  $P_i$  is unique in paths  $p_k$  ordering, e.g.,

$$\{P_i\} = \{\{p_1, p_2, p_3\}, \{p_1, p_3, p_2\}, \{p_2, p_1, p_3\}, \{p_2, p_3, p_1\}, \{p_3, p_1, p_2\}, \{p_3, p_2, p_1\}\}.$$

On any iteration  $P_i$ , the path  $p_k$  is consequently examined for feasible flow  $F_i = \sum\{f(p_k)\}_i$  until no more paths  $p_k$  exist. Among the iterations  $\{F_i\}$ , one or more best results  $\{F_{\max}\} \subset \{F_i\}$  are to be taken as final product. The power of the set  $\{F_i\}$  steamily grows with the increase of vertex's quantity V.

To reduce this task, apply the known heuristic approach on the base of Pontryagin maximum

principle. Let two extremum necessary conditions for each step 'k' of the current iteration 'i':

- 1) "max-flow-try", i.e.  $f(pk)=max$ ;
- 2) "shortest-path-first", i.e.  $pk \leq pk+1$  in hops.

The first condition is used in common FFA-based algorithms of MFP solution on the ST-planar directed graphs; this ensures convergence of the solution process.

The second condition is used in FFA-based Edmonds-Karp algorithm of MFP to reduce computational complexity. Figure 6 helps to see the necessity of the 'shortest-path-first' condition to achieve the maximum flow in ST-planar free-oriented graph.

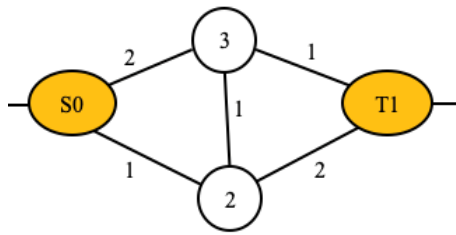


Figure 6: ST-planar graph case with critical first path.

It is clear, that the feasible flow from the source S0 to the target T1 in Figure 6 is  $F_{max} = f_1(S0-2-T1) + f_2(S0-3-T1) + f_3(S0-3-2-T1) = (1+1+1) = 3$ , whereas the residual graph is null. However, in case of violation the second necessary condition ('shortest-path-first'), e.g. if begin with the 3-hop path  $p_1=(S0-2-3-T1)$  instead 2-hop path (S0-2-T1), we obtain less feasible flow  $F=1 < F_{max}=3$ , and non-null residual graph  $G_R$ , whereas no more paths from S0 to T1 exist:

$$F=f_1(S0-2-3-T1)=1 < F_{max}=3;$$

$$G_R=\{(3, S0)=2; (3, T1)=2\};$$

vertices S0 and T1 are isolated.

Thus, violating the second necessary condition ('shortest-path-first') may result in false output product (at least one case shown above); therefore, it is always needed, i.e. is necessary condition.

Let estimate the complexity reduction of the inverse MFP-task by the 'shortest-path-first' condition provision. The total number of different h-hop paths in ST-planar graph  $G(V)$  with V vertices can be calculated with Figure 3:

$$NPV=1+\sum(V-h), h \in 2, 3, \dots, V-1.$$

In particular case of  $V=6$ , it is

$$NP6=1+(6-2)+(6-3)+(6-4)+(6-5)=11.$$

It is easy to show, that in general case, the number of h-hop paths in ST-graph with V vertices is polynomial function

$$N_{PV} = 1+\sum(V-2, V-3, \dots, 1) = 1+0.5(V-2)(V-1).$$

So, each set  $P_i=\{p_k\}_i$  has  $N_{PV}$  paths number.

Let  $\{P_i\}$  be the class of all the sets  $P_i$ . According to the known combinatorial formulas, the number of sets  $P_i$  in the class  $\{P_i\}$  equals the number of permutations of the paths  $p_k$ :

$$|\{P_i\}|=1 \cdot 2 \cdot 3 \cdot \dots \cdot NPV=(NPV)!.$$

For instance, if  $V=6$ , then  $|\{P_i\}|=11!=39\ 916\ 800$ .

It is rather clear, that among all the  $P_i$  sets within the full  $\{P_i\}$  collection, the only one set  $P \in \{P_i\}$  may satisfy the "shortest-path-first" condition.

Thus, by applying Pontryagin maximum principle along with two heuristic necessary conditions ("max-flow-try" and "shortest-path-first"), the inverse combinatorial Maxflow task (IMT) on the bi-polar free-oriented ST-planar network graph can be reduced from the none-deterministic polynomial complexity NPC, with  $\{(N_{PV})!\}$  iterations for scanning sets of paths  $\{P_i\}$ , to PC-task with computation the flow distribution over a single set of paths  $P=\{p_k\}$ , that includes the polynomial number of paths  $N_{PV} = 1+0.5(V-2)(V-1)$ .

Now, we get the answer, what ML training sequence TSS is sufficient for testing the IMT-algorithms on the FWG network graph. It is the set of paths  $N_{PV}$ , constructed by Pontryagin maximum principle with two extremum necessary conditions.

## 7 CONCLUSIONS

The main scientific result of the work is formalization of the inverse and direct Maxflow tasks on the free-oriented weighted ST-planar network graph. This allows to use the dynamic reconfiguration of the modern telecommunication channels in digital network-flow optimization, in order to increase the overall networks performance. Within the scope of this result, the following is obtained.

The Maxflow problem state of the art is analyzed in Section 2. It shows, that most publications on the Maxflow problem explore the so called "logistic models" of transportation system in the form of directed bipolar ST-graph. Among them, recent researches on artificial intelligence and machine learning techniques have been increasingly used to approach the Maxflow problem.

It is concluded, that conventional logistic Maxflow models are not enough adequate to modern



telecoms, and therefore, hamper to fully benefit the SDN-reconfigurability. Known algorithms on multi-pole free-oriented graphs are solely limited by the 6-node graph case. Because of that, further researches on Maxflow methods in telecoms on the free-oriented graph model needed.

To advance the Maxflow problem study in data networks with reconfigurable channels, related objectives have been set in Section 3. Section 4 formalizes the inverse and direct Maxflow tasks on the free-oriented bi-polar ST-planar network graph in terms of inverse Maxflow task testing and machine learning of artificial intelligence models. Section 5 studies the direct Maxflow task as a first part of the whole Maxflow problem in tensor form; an algorithm of testing samples set calculation is constructed.

The inverse Maxflow problem has been analyzed in Section 6 as a discrete optimization task on the Pontryagin maximum principle with two necessary extremum conditions: ‘max-flow-try’ for the flow-path scanning, and “shortest-path-first” for augmenting paths searching. The related algorithm is reduced to a single iteration of paths tensor analysis with polynomial paths number.

In general, unlike the known approaches to product flow maximization on logistic system model, a novel method introduced for digital flow optimization in software defined networks with dynamically reconfigurable channels. Along with the total maximal flow, this method provides the maximal flow distribution over the network structure. The direct Maxflow formalism also enables the Maxflow algorithms testing and machine learning of artificial intelligence models.

## REFERENCES

- [1] L.R. Ford and D.R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, 1956, pp. 399-404. [Online]. Available: <https://www.semanticscholar.org/paper/Maximal-Flow-Through-a-Network-Ford-Fulkerson/794b01b2513f3610cb151ecfd07e9dc0ea7e1f3>.
- [2] Y. Dinitz, "Algorithm for solution of a problem of maximum flow in a network with power estimation," 1970. [Online]. Available: [https://www.researchgate.net/publication/228057696\\_Algorithm\\_for\\_Solution\\_of\\_a\\_Problem\\_of\\_Maximum\\_Flow\\_in\\_Networks\\_with\\_Power\\_Estimation](https://www.researchgate.net/publication/228057696_Algorithm_for_Solution_of_a_Problem_of_Maximum_Flow_in_Networks_with_Power_Estimation).
- [3] J. Edmonds and R. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," *Journal of the Association for Computing Machinery*, vol. 19, no. 2, 1972, pp. 248-264.
- [4] J.A. Bondy, et al., "Graph theory with applications," Macmillan Press Ltd, GB, 1976. [Online]. Available: <https://www.iro.umontreal.ca/~hahn/ift3545/gtwa.pdf>.
- [5] A.V. Goldberg and R.E. Tarjan, "A new approach to the maximum flow problem," *Proceedings of the eighteenth annual ACM symposium on Theory of computing – STOC '86*, 1986, pp. 136-146.
- [6] A. V. Goldberg and S. Rao, "Beyond the flow decomposition barrier," *J. ACM*, vol. 45, no. 5, 1998, pp. 783-797.
- [7] D. Papp, "The Goldberg-Rao algorithm for the maximum flow problem," COS 528 class notes, 2006, 5 p. [Online]. Available: <https://www.cs.princeton.edu/courses/archive/fall06/cos528/handouts/Goldberg-Rao.pdf>.
- [8] P. Christiano, et al., "Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs," *Computer Science, Data Structures and Algorithms*, 2010. [Online]. Available: <https://www.semanticscholar.org/reader/af1afe809c106ed2618fc2ae14b696f672fc4bc1>.
- [9] M.A. Rajouh, "Osobenosti primeneniya principa maksimuma Pontryagina," BNTU, 2013. [Online]. Available: [https://rep.bntu.by/bitstream/handle/data/5527/Osobennosti\\_primneneniya.pdf?sequence=1&isAllowed=y](https://rep.bntu.by/bitstream/handle/data/5527/Osobennosti_primneneniya.pdf?sequence=1&isAllowed=y).
- [10] D.P. Williamson, "Network flow algorithms," Cornell University, 2019. [Online]. Available: <https://www.networkflowalgs.com/book.pdf>.
- [11] O.V. Tykhonova, et al., "The Max-Flow Problem Statement on the Three-Pole Open Network Graph," 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), Lviv, Ukraine, 2019, pp. 209-212, doi: 10.1109/AICT.2019.8847745. [Online]. Available: <https://ieeexplore.ieee.org/document/8847745/metrics#metrics>.
- [12] O.V. Tykhonova, "Conveyor-modular method of multimedia flows integration with delay control in packet based telecommunication network," PhD Dissertation, O.S.Popov ONAT, Odessa, 2019.
- [13] R. Haese et al., "Algorithms and Complexity for the Almost Equal Maximum Flow Problem," *Operations Research Proceedings 2019*. Springer, Cham, 2020, pp. 323-329.
- [14] A. Alzaben, et al., "End-to-End Routing in SDN Controllers Using Max-Flow Min-Cut Route Selection Algorithm," 23rd International Conference on Advanced Communication Technology (ICTACT), 2021, pp. 461-467.
- [15] H. Liu, et al., "Optimization of a logistics transportation network based on a genetic algorithm," *Hindawi Mobile Information Systems*, vol. 2022, 2022, pp. 167-176. [Online]. Available: <https://doi.org/10.1155/2022/1271488>.
- [16] O. Cruz-Mejía and A.N. Letchford, "A survey on exact algorithms for the maximum flow and minimum-cost flow problems," *Networks*, vol. 82, no. 2, 2023. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/net.22169>.

- [17] Y. Bengio, et al., "Machine learning for combinatorial optimization: A methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, 2021, pp. 405-421.
- [18] B. Mor, et al., "Heuristic algorithms for solving a set of NP-hard single-machine scheduling problems with resource-dependent processing times," *Computers & Industrial Engineering*, vol. 153, 2021.
- [19] W. Jiang, "Graph-based Deep Learning for Communication Networks: A Survey," *Computer Communications*, vol. 185, 2022, pp. 40-54. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0140366421004874?via%3Dihub>.
- [20] N. Orkun Baycik, "Machine learning based approaches to solve the maximum flow network interdiction problem," *Computers & Industrial Engineering*, vol. 167, 2022.
- [21] T. Tawanda, et al., "An intelligent node labelling maximum flow algorithm," *Int J Syst Assur Eng Manag* 14, pp. 1276-1284, 2023. [Online]. Available: <https://doi.org/10.1007/s13198-023-01930-3>.
- [22] "Graph theory," *Britannica*, 2023. [Online]. Available: <https://www.britannica.com/topic/graph-theory>.
- [23] L. Wagner, "Complete Python Tutorial for Absolute Beginners," 2023.
- [24] A. Kuronya, "Introduction to topology," 2010, 102 p.
- [25] Matrix calculator. [Online]. Available: <https://matrixcalc.org>.