

DOI: <https://doi.org/10.36910/6775-2524-0560-2025-59-41>

УДК 621.391.8:004.382

Syvolovskiy Iliia¹, PhD student<https://orcid.org/0000-0002-4592-0965>Komar Oleksii², PhD, Associate Professor<https://orcid.org/0009-0002-2994-6556>¹Ukrainian State University of Railway Transport, Kharkiv, Ukraine²National Aviation University, Kyiv, Ukraine

A METHOD OF MULTICRITERIA DATA STREAM DISTRIBUTION IN TELECOMMUNICATION NETWORKS BASED ON AN EVOLUTIONARY APPROACH

Syvolovskiy I., Komar O. A method of multicriteria data stream distribution in telecommunication networks based on an evolutionary approach. The article presents a method of multicriteria decision-making for the distribution of data streams in telecommunication systems, developed on the basis of the modified genetic algorithm NSGA-III. The proposed model takes into account the dynamic nature of the load, resource constraints, the possibility of delegating tasks between clusters, and predicting peak traffic surges. The problem is formalized as a generalized scheduling problem with a set of criteria, including minimizing the use of node resources, load balancing, and reducing the number of delegated streams. The architecture of the system with the logic of stream processing and interaction of cluster coordinators is described. The developed algorithm includes adaptive updating of reference directions, hybrid ranking taking into account the probability of overload, and dynamic adjustment of the mutation rate according to the predicted load. The effectiveness of the proposed approach is confirmed by calculating the fitness function and analyzing the resulting Pareto fronts. It is substantiated that the method allows maintaining high flexibility and accuracy of data stream (load) distribution in the variable environment of telecommunication networks.

Keywords: telecommunication systems, evolutionary approach, genetic algorithm, encoding, stream, node, traffic, workload, resource intensity, optimization, algorithm implementation, distribution of data streams, solution, scheduling problem.

Сиволовський І. М., Комар О. М. Метод багатокритеріального розподілу інформаційних потоків у телекомунікаційних мережах на основі еволюційного підходу. У статті представлено комплексне дослідження, присвячене розробці адаптивного методу реконструкції сигналів у динамічних середовищах. Запропонований метод базується на використанні модифікованих рядів Вольєрра з часовими обмеженнями, де внесок ядер обмежується локальними часовими вікнами, визначеними за допомогою згладжувальної Гаусової функції. Такий підхід дозволяє подолати обмеження традиційних спектральних методів, які внаслідок згладжувального ефекту не здатні точно відтворювати швидкоплинні або імпульсні особливості сигналу. Для виявлення критичних ділянок сигналу, а саме областей з різкими змінами або локальними аномаліями, в роботі введено індикатор нестабільності, що дозволяє здійснювати вибірку активацію часово обмеженої моделі лише в нестійких зонах. У стабільних ділянках сигналу реконструкція виконується з використанням частотної моделі, що забезпечує ефективне використання обчислювальних ресурсів. За результатами експериментів отримано зростання коефіцієнта локальної узгодженості (ALC) в діапазоні 10–14% в залежності від просторової локалізації критичних точок та інтенсивності часових змін сигналу, а також зменшення середньоквадратичної похибки (MSE) на 12–18% у порівнянні з традиційними методами частотної реконструкції. Отримані результати підтверджують ефективність запропонованого методу у задачах обробки сигналів для когнітивних телекомунікаційних систем в умовах складного заводового середовища.

Ключові слова: телекомунікаційні системи, еволюційний підхід, генетичний алгоритм, кодування, потік, вузол, трафік, навантаження, ресурсоемність, оптимізація, алгоритм реалізації, розподіл інформаційних потоків, рішення, задача розкладу.

Statement of a scientific problem. In distributed telecommunication systems, efficient processing of data streams is complicated by limited resources, uneven cluster load, traffic spikes, and the need for prompt load delegation. Such systems are characterized by dynamic nature and the presence of many conflicting criteria, which requires finding compromise solutions in real time. To solve this scientific and practical problem, taking into account modern conceptual approaches to multi-objective evolutionary optimization [1-17], a multi-criteria decision-making method for the distribution of data streams is proposed, which involves:

- representing decisions in the form of chromosomes with the assignment of streams to nodes;
- evaluating decisions by a fitness function that takes into account the number of delegations, load balance, and risk of overload;
- adaptive response to changes in traffic by dynamically updating parameters.

The method is versatile and can be implemented on the basis of any multicriteria genetic algorithm. In this study, NSGA-III was chosen because it provides efficient coverage of the Pareto space with a large number of criteria and allows for easy integration of adaptive mechanisms.

The proposed method generates a stable solution under unstable load conditions, provides flexible control over system resources, and can be used for dynamic real-time stream control.

Research analysis. An analysis of current research in the field of multicriteria evolutionary optimization and genetic algorithms in distributed computing environments shows that the problem of efficient stream management under dynamic load remains relevant. The works [1, 8, 9, 11, 17] consider the use of genetic algorithms for routing, coverage, and resource allocation in bandwidth-limited environments, but without taking into account multicriteria and adaptability to changes.

Publications [4-7, 12, 15, 16] focus on the development of NSGA-III and related algorithms. They propose improvements to the approaches for sorting and generating reference points, but the issues of adaptation to variable load and incorporation of predictions remain insufficiently covered. Studies [2, 10, 13] analyze the limitations of metaheuristic methods, and works [3, 14] demonstrate the use of multicriteria optimization for service placement, but do not take into account the specifics of traffic in telecommunications networks.

Thus, the obtained reference scientific and practical results create the basis for further research, in particular in the direction of developing adaptive multicriteria methods for distributing data streams in telecommunication networks, taking into account load spikes and real-time delegation mechanisms.

The purpose of the work. The aim of the study is to develop a method for multicriteria distribution of data streams in telecommunication networks based on the modified genetic algorithm NSGA-III, taking into account the dynamic load, resource constraints and predicted traffic spikes.

Presentation of the main material and substantiation of the obtained research results.

The model of the system in which the proposed method is implemented describes a distributed environment for processing data streams [1,8], and is based on an architecture specially developed for this study, consisting of clusters of computing nodes that can perform a task of the same type. Each cluster contains heterogeneous computing resources and has a coordinator that centrally manages the assignment of tasks to cluster nodes. In case of resource shortage, a mechanism is provided for delegating streams to neighboring clusters or a global supervisor. Data streams enter the system with different rates, may differ in type, resource intensity, and have the property of sudden spikes (increase in resource intensity for a certain period of time). The generalized architecture of the proposed system is shown in Fig. 1.

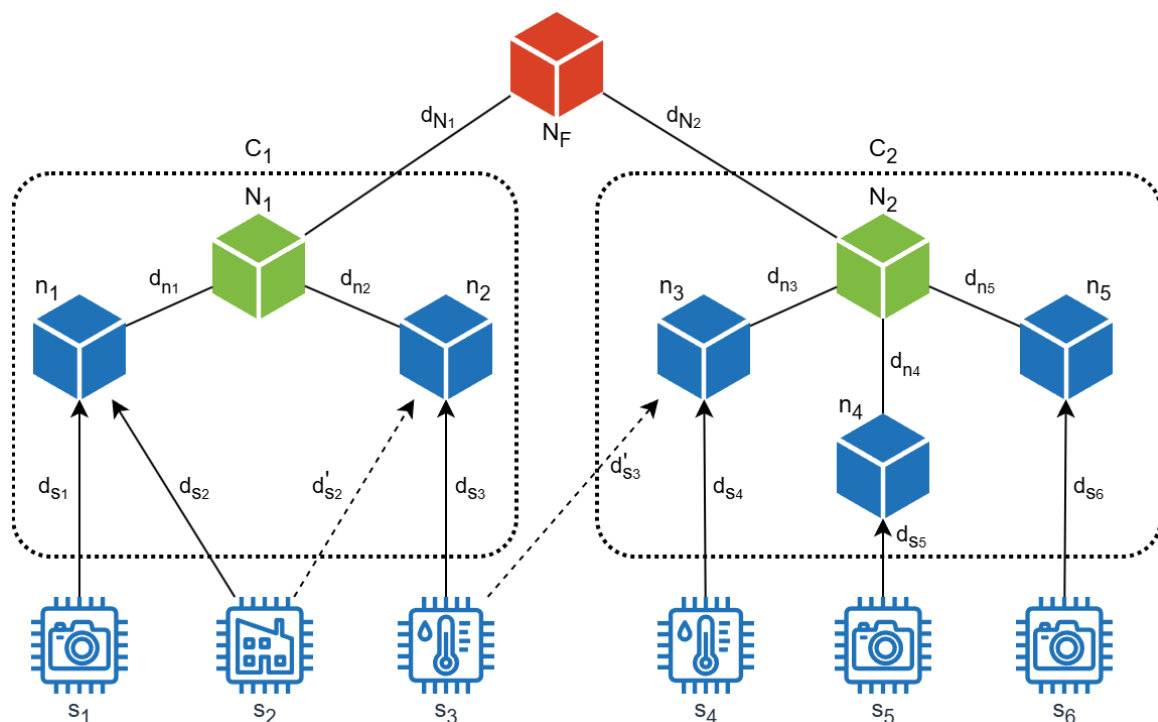


Fig.1 – System architecture with cluster coordinators and delegation mechanisms

Figure shows that the architecture of the proposed system is represented by clusters (C_1, C_2), each of which consists of computing nodes ($n_1 - n_5$) that perform tasks of the same type but differ in computing capabilities (CPU, RAM, Storage). Each cluster has a coordinator (N_1, N_2), that centrally manages the

distribution of incoming data streams ($d_{S1} - d_{S6}$). In case of resource shortage, a cluster can delegate the data stream to another cluster node ($d_{S2} \rightarrow d'_{S2}$), another cluster ($d_{S3} \rightarrow d'_{S3}$), or a global supervisor (N_F). This architecture forms an adaptive distributed system that can flexibly respond to changes in load, including spikes in stream resource intensity.

The diagram shows computing nodes that physically process streams in blue, cluster coordinators that decide on thread distribution in green, and a global supervisor (N_F) to which streams can be transferred if the cluster cannot process them in red.

At the bottom of the figure are icons that represent typical sources of data streams (s_1-s_6). Each of them illustrates an example of a device or a type of task that generates the corresponding stream:

- the house icon represents streams from smart home devices or the industrial IoT ecosystem;
- the camera icon represents video streams or individual image processing tasks;
- the thermometer icon represents streams from temperature sensors or similar environmental monitoring devices.

The proposed architecture forms an adaptive distributed system capable of responding flexibly to load changes, including sudden spikes in resource intensity, which are modeled as statistical events with a probability that increases over time.

Taking into account the objectives of this study, as well as the logic of the distribution of data streams between nodes in the described environment, the model can be presented as a type of multi-criteria scheduling problem [6,8,14]. It includes elements of the placement problem, but is extended with specific constraints that reflect the features and requirements of this study:

- processing of tasks of only a fixed type within each cluster;
- limited computing resources (CPU, RAM, Storage);
- risk of overloading during sudden load surges;
- the need to support flexible delegation in case of resource shortages.

The main characteristics of the system that are taken into account when modeling and solving the optimal distribution problem are shown in Table 1.

Table 1 – Main features of the system

Element	Description
Types of tasks	There are several fixed types of tasks (for example, 1-4); each cluster processes tasks of only one type
Data streams	Each stream has: task type, resource intensity (CPU, RAM, Storage), the possibility of short-term spikes in resource intensity
Processing nodes	Within the cluster; heterogeneous in terms of resources; process only streams of one type of task
Clusters	Combine nodes of the same type; each has a coordinator who performs centralized planning
Coordinators	Assign streams to local nodes based on available resources; do not have information about the detailed state of other clusters
Delegation	Only possible in case of overload: to neighboring clusters or to the global supervisor
Load spikes	Occur with a probability that grows exponentially; last for an interval τ and lead to a temporary increase in resource requirements
Stream restrictions	Assignment of each stream: only one node or delegation; control of resource overload; minimization of delegation
Optimization	Utilization of resources, number of delegations, risk of overloading during spikes,

criteria	load balance
----------	--------------

Table 2 presents the author's formalization of the main parameters, structural components, and variables used in the proposed method of multicriteria distribution of data streams.

Table 2 – Parameters and notation of indicators

Designation	Value
C	The set of all clusters $\{C_1, C_2, \dots, C_i\}$ that process tasks of type T_k
n_{c_i}	Node in cluster C_i that processes tasks of type T_k
$S_{c_i}^n$	The set of all computing nodes within the cluster C_i : $\{n_1, n_2, \dots, n_j\}$
N_{c_i}	The main cluster node (coordinator) that distributes the load
N_F	A node that acts as a global system supervisor
D	The set of all input data streams $\{d_1, d_2, \dots, d_m\}$
NB_C	A set of neighboring clusters: $NB_C = \{C'_1, \dots, C'_k\}$
EX_C	External resources of cluster C : $EX_C = \{NB_C, N_F\}$
$R_{n_{c_i}}$	The set of resources of a node in the cluster C_i : $(CPU_{n_{c_i}}, RAM_{n_{c_i}}, STG_{n_{c_i}})$
R_{d_m}	Resource costs (cpu_m, ram_m, stg_m) for processing the data stream d_m
τ	The duration of a possible load spike on the stream
λ	Exponential distribution parameter for estimating the probability of a spike
σ	Load multiplier during the spike. Shows how much additional load the stream consumes during a spike $(\sigma_{cpu}, \sigma_{ram}, \sigma_{stg})$
$x_{m,n_{c_i}}$	Decision variable. It has a value of 1 when assigning a stream d_m to node n_{c_i} within the cluster
y_{m,EX_C}	Decision variable. It has a value of 1 when assigning a stream d_m to a resource from outside EX_C (another cluster or supervisor)
X_m	Generalized decision variable - assignment of a thread d_m to a resource (node, another cluster or supervisor)
F	The objective function vector $F = [f_1, f_2, f_3, f_4]$, which includes optimization criteria: minimization of resource utilization f_1 , number of delegations f_2 , overload during spikes f_3 and load imbalance f_4
w_{f_a}	Weighting coefficients for each criterion in the function F

To implement the proposed method of multi-criteria distribution of data streams, a system of objective functions f_1, f_2, f_3, f_4 is used to evaluate each assignment option [6,7,13]. In this study, each

possible distribution (potential solution) is evaluated using the vector objective function $F = f_1, f_2, f_3, f_4$, which takes into account both the current state of the computing environment and the potential risks of its overload in the future.

The developed system of criteria within the framework of the proposed method allows achieving the following scientific goals.

1. Assessment of resource utilization efficiency. This aspect is described by the criterion f_1 – minimizing the total use of local resources (processor, RAM, storage). It is calculated by the formula (see Table 2 for the designation of indicators):

$$\min f_1 = \sum_{n \in S_C^n} \left(\frac{1}{CPU_n} \sum_{d \in D} x_{d,n} \cdot cpu_d + \frac{1}{RAM_n} \sum_{d \in D} x_{d,n} \cdot ram_d + \frac{1}{STG_n} \sum_{d \in D} x_{d,n} \cdot stg_d \right). \quad (1)$$

2. Assessment of the need to delegate streams. Meets criterion f_2 – minimizing the number of streams that have been delegated to external resources (neighboring clusters or a supervisor):

$$\min f_2 = \sum_{i=1}^m y_{i,EX}. \quad (2)$$

3. Predicting the risk of resource overload in the case of spikes. Criterion f_3 – minimizing the expected overload of resources during a load spike τ , taking into account the probability of its occurrence:

$$\min f_3 = \sum_{n \in S_C^n} (CPU_n^{over} + RAM_n^{over} + STO_n^{over}). \quad (3)$$

The expected resource overload for each node n is calculated using the formulas:

$$CPU_n^{over} = \max \left(0, \sum_{d \in D} (P_d^{surge} \cdot x_{d,n} \cdot \sigma_{cpu} \cdot cpu_i) - CPU_n \right), \quad (4)$$

$$RAM_n^{over} = \max \left(0, \sum_{d \in D} (P_d^{surge} \cdot x_{d,n} \cdot \sigma_{ram} \cdot ram_i) - RAM_n \right), \quad (5)$$

$$STO_n^{over} = \max \left(0, \sum_{d \in D} (P_d^{surge} \cdot x_{d,n} \cdot \sigma_{stg} \cdot sto_i) - STO_n \right). \quad (6)$$

Where the probability of a load spike is calculated as:

$$p_{surge} = 1 - e^{-\lambda\tau}. \quad (7)$$

4. Ensuring a balanced distribution of streams between cluster nodes. Evaluated by the f_4 criterion, it minimizes the variance (Var) of the load per node (for CPU, RAM, storage), which ensures a balanced distribution of streams:

$$\min f_4 = Var^{CPU} + Var^{RAM} + Var^{STG}, \quad (8)$$

$$\begin{cases} Var^{CPU} = Var(CPU_1^{load}, CPU_2^{load}, \dots, CPU_j^{load}), \\ Var^{RAM} = Var(RAM_1^{load}, RAM_2^{load}, \dots, RAM_j^{load}), \\ Var^{STG} = Var(STG_1^{load}, STG_2^{load}, \dots, STG_j^{load}), \end{cases} \quad (9)$$

$$\begin{cases} CPU_n^{load} = \frac{1}{CPU_n} \sum_{d \in D} x_{d,n} \cdot cpu_d, \\ RAM_n^{load} = \frac{1}{RAM_n} \sum_{d \in D} x_{d,n} \cdot ram_d, \\ STG_n^{load} = \frac{1}{STG_n} \sum_{d \in D} x_{d,n} \cdot stg_d. \end{cases} \quad (10)$$

The described optimization criteria are applied in the presence of a system of constraints.

1. Unambiguous assignment constraint (each stream must be either processed locally or delegated):

$$X_d = \sum_{n \in S_C^n} x_{d,n} + y_{d,EX} = 1, \forall d \in D. \quad (11)$$

2. Limiting the amount of node resources (each node is not allowed to exceed the available resources):

$$\sum_{d \in D} x_{d,n} \cdot cpu_d \leq CPU_n, \forall n \in S_C^n, \quad (12)$$

$$\sum_{d \in D} x_{d,n} \cdot ram_d \leq RAM_n, \forall n \in S_C^n, \quad (13)$$

$$\sum_{d \in D} x_{d,n} \cdot sto_d \leq STO_n, \forall n \in S_C^n. \quad (14)$$

3. Compatibility of task types (each node can only process tasks of a certain type - this is inherent in the cluster structure and is ensured at the stage of chromosome initialization).

Given these limitations, each potential solution is evaluated using an additive fitness function:

$$F = w_1 \cdot f_1 + w_2 \cdot f_2 + w_3 \cdot f_3 + w_4 \cdot f_4, \quad (15)$$

To implement the proposed method, a genetic representation of potential solutions in the form of a chromosome is used. Each chromosome encodes one of the possible options for assigning data streams to available resources, taking into account both local nodes and the possibility of delegation.

There are two ways to indicate whether a stream belongs to a node in a chromosome: binary and integer [14].

1. Binary encoding involves creating a set of bits for each stream. The size of such a set is equal to the number of all possible placement directions (all cluster nodes, supervisor, neighboring clusters).

2. The integer encoding uses only one array of integers, where the absolute value of the number is the resource identifier and the sign is its type:

- if the number is greater than 0, it is the identifier of the local node within the cluster;
- if the number is equal to 0, the stream should be allocated to the supervisor node;
- if the number is less than 0, it is the identifier of the cluster to which the stream should be allocated.

If complex cluster and node identifiers are used, this approach can be combined with a mapping table, if necessary, to restore the exact correspondence between the identifier and the resource.

Let's estimate the amount of memory required to represent a chromosome in each of the encoding methods, taking into account the parameters defined for the tasks of this study (Table 3).

The comparison of approaches shown in Table 4 allows us to conclude that the integer approach to chromosome encoding is more efficient for the tasks of this study. Its use provides a significant reduction in memory size, higher scalability, and avoids restrictions on the maximum number of nodes, since there is no need to additionally control whether the stream belongs to internal or external resources.

Table 3 – Memory evaluation for chromosome representation

Parameter	Value
Number of streams S	1000
Number of internal nodes N	16
Number of neighboring clusters N_b	7
Presence of a supervisor N_F	1
The size of the integer cluster identifier	8 bit (1 byte)
Size in memory (binary encoding) $Size_{bin}$	$1000 \times (16+7+1) = 24000$ byte
Size in memory (integer encoding) $Size_{int}$	$1000 \times 8 = 8000$ byte

Table 4 – Evaluating the efficiency of binary and integer encoding

Feature	Binary encoding	Integer encoding
Size in memory	High	Low
Decoding complexity	Simple	Requires mapping
Flexibility of scaling	Limited by the number of nodes	Higher – does not depend on a fixed N
Search speed	Fast bit recognition	Possibly slower, depending on implementation
Versatility	Not suitable for hybrid structures	Works effectively with hybrid systems

The structure of a chromosome using integer coding is shown in Fig. 2.

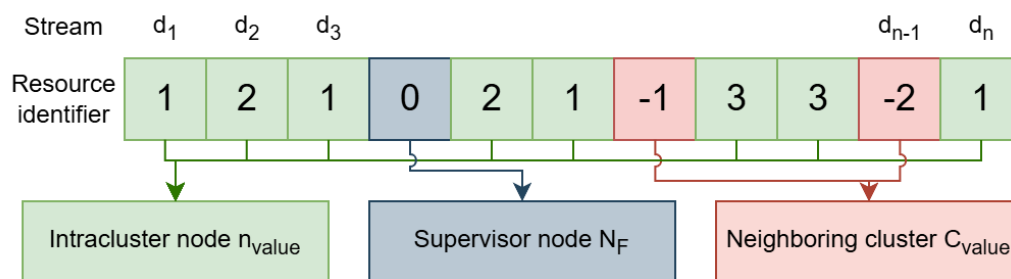


Fig. 2 – Representation of a chromosome using an integer approach

Based on the given chromosome structure, a modification of the NSGA-III algorithm is proposed to implement the method of multicriteria distribution of data streams. The updated block diagram of the method is shown in Fig. 3. The stages that have been improved are highlighted in green.

The main stages of the modified genetic algorithm NSGA-III include the following.

Stage 1. Initialization of parameters.

At the initial stage, the basic parameters of the genetic algorithm are determined: the number of solutions in the population L_p , the maximum number of generations G_{max} , crossover coefficients P_c and P_m , the number of reference points H , the initial directions r_0 , adaptation coefficients a_j , the threshold for improving the value of the fitness function θ .

Stage 2. Generation of the initial population.

The initial population P_0 is not generated randomly, but using heuristics (e.g., First-Fit, Best-Fit, or Least-Loaded methods). This improves the quality and validity of the initial solutions, and also provides for the reservation of some nodes as a buffer in case of future spikes.

Stage 3. Generation of reference points.

Initial reference points are created in the solution space, which are used to orient the algorithm when searching for Pareto-optimal solutions.

Step 4. Calculation of the objective function (fitness function) taking into account the load prediction (improvement).

This stage is realized by estimating the probable load spikes. For each solution (chromosome), the values of the objective functions are calculated, taking into account not only the current but also the potential future load spikes.

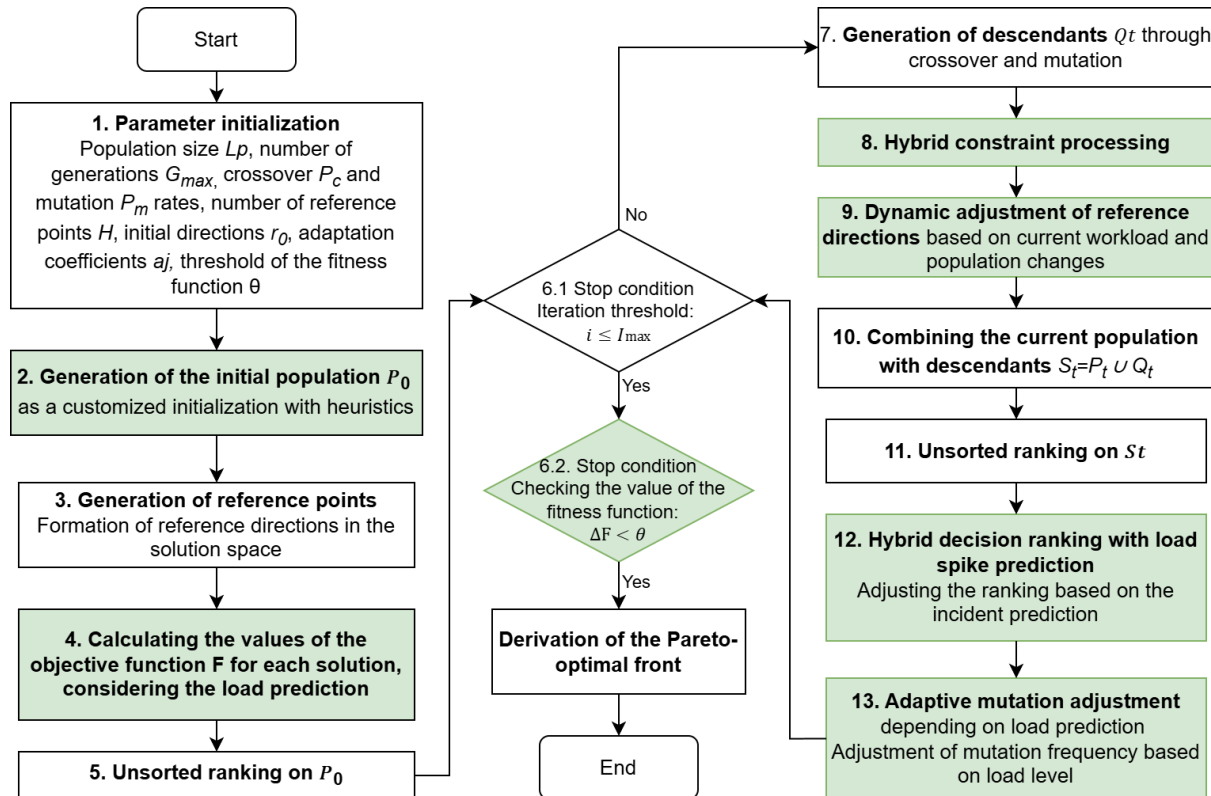


Fig.3 – Flowchart of the modified NSGA-III algorithm

Stage 5. Unsorted ranking of the initial population.

Ranking of P_0 is performed taking into account the multicriteria evaluation, and a decision front without dominance is formed (unsorted ranking).

Stage 6. Checking the termination conditions.

The conditions for completing the algorithm are checked:

– reaching the maximum number of generations $i \leq I_{max}$ – the number of iterations does not exceed the maximum limit; -

– the improvement in the value of the fitness function between the last and current iteration is less than the threshold $\Delta F < \theta$.

If any of the conditions is met, the algorithm stops and the found Pareto-optimal front is displayed.

Stage 7. Generation of descendants.

New potential solutions (descendants) are created using genetic operators Q_t – crossover and mutation.

Step 8. Hybrid constraint processing (improvement).

Correction of unacceptable solutions is performed (repair operator). Adaptive penalty coefficients $penalty_{adapt}$ are introduced.

Step 9. Dynamic adjustment of reference directions (improvement).

The reference directions are updated adaptively according to load changes, where r_t^{adapt} is the adaptive reference directions at iteration t .

Step 10. Combining the current population with the offspring.

The current population and the newly generated offspring are combined for further ranking and selection: $S_t = P_t \cup Q_t$

Step 11. Unsorted ranking of the combined population.

The ranking of the combined population S_t is performed, and a new decision front without dominance is formed.

Stage 12. Hybrid ranking of solutions with spike load prediction (improvement).

At this stage, the ranking of solutions is further refined, taking into account not only the current state but also the expected spike load $rank_t^{hyb}$. This allows you to more accurately prioritize solutions based on the risks of future spikes.

Step 13. Adaptive mutation tuning with load prediction (improvement).

At this stage, the mutation rate is adjusted depending on the P_m^{adapt} load prediction, which allows the algorithm to effectively avoid overloads and flexibly respond to changes in the characteristics of input streams.

After performing the adaptive mutation adjustment, the algorithm terminates the current iteration. Next, the algorithm proceeds to check the termination conditions (Stage 6), where it analyzes whether the maximum number of generations I_{max} has been reached or whether the fitness function ΔF has improved between the last and current iteration. If at least one of these conditions is met, the algorithm terminates and the final Pareto-optimal front is generated, which contains the optimal solutions for the distribution of streams, taking into account the defined criteria. If the termination conditions are not met, the next iteration is started (returning to Stage 7) with the subsequent generation of new descendants, and the entire cycle is repeated until the termination conditions are met.

Evaluation of the operational capability of the algorithm implementing the developed method by simulation modeling in Python showed that this method improves system performance in the range of 10-15%.

The proposed modification of NSGA-III provides: high adaptability of the algorithm to load changes in the conditions of dynamic functioning of telecommunication networks: efficient resource balancing and spike load prediction in a distributed environment; fast convergence to high-quality Pareto-optimal solutions, taking into account the specifics of data streams processing and resource constraints.

Conclusions and prospects for further research. The article proposes a method of multi-criteria distribution of data streams in telecommunication networks using the modified NSGA-III genetic algorithm. A mathematical model has been developed that takes into account the limited computing resources, the need for flexible delegation of streams, and the risks of overload during spikes in load. The solution is implemented by genetically encoding the distribution of streams in the form of chromosomes using an integer approach, which is expected to provide high memory efficiency, scalability, and versatility of use in heterogeneous distributed environments.

The developed method allows for adaptive planning of incoming data streams in real time, taking into account load changes and spike load predictions, which can increase network performance by 10-15%. Improvements to the NSGA-III algorithm have improved the quality of Pareto-optimal solutions, flexibility in handling constraints, and efficient load balancing between nodes.

Prospects for further research include: integration of traffic prediction tools using machine learning methods, development of energy saving mechanisms for stream distribution, and implementation of the proposed method in a simulation environment for deeper experimental testing.

References

1. Abdallah W., Val T. (2020). Genetic-Voronoi algorithm for coverage of IoT data collection networks. *30th International Conference on Computer Theory and Applications (ICCTA)*, Alexandria, Egypt, 2020. PP. 16-22. DOI: <https://doi.org/10.48550/arXiv.2202.13735>
2. Bernardino R., Paiao A. (2018). Metaheuristics based on decision hierarchies for the traveling purchaser problem. *International Transactions on Operational Research*, 25(4), 2018. PP. 1269-1295. DOI: <https://doi.org/10.1111/itor.12330>
3. Blank J., Deb K. (2020) Pymoo: Multi-Objective Optimization. *IEEE Access*, Vol. 8, 2020. PP. 89497-89509. DOI: <https://doi.org/10.1109/ACCESS.2020.2990567>
4. Blank J., Deb K., Dhebar Y., Bandaru S., Seada H. (2012). Generating well-spaced points on a unit simplex for evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1), 2021. PP. 48-60. DOI: <https://doi.org/10.1109/TEVC.2020.2992387>

5. Cheng R., Jin Y., Olhofer M., Sendhoff B. (2016). A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5), 2016. PP. 773-791. DOI: <https://doi.org/10.1109/TEVC.2016.2519378>
6. Deb K., Abouhawwash M. (2016). An optimality theory-based proximity measure for set-based multiobjective optimization. *IEEE Trans. Evolutionary Computation*, 20(4):515–528, 2016. DOI: <https://doi.org/10.1109/TEVC.2015.2483590>
7. Deb K., Jain H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4), 2014. PP. 577-601. DOI: <https://doi.org/10.1109/TEVC.2013.2281535>
8. Guerrero C., Lera I. Juiz C. (2018). Genetic algorithm for multi-objective optimization of container allocation in cloud architectures. *Journal of Grid Computing*, 16-1, 2018. PP. 113-135. DOI: <https://doi.org/10.1007/s10723-017-9419-x>
9. Lian Y., Peng P., Jiang K., Xu W. (2024). Multi-objective compression for CNNs via evolutionary algorithm. *Information Sciences*, Volume 661, March 2024. PP. 120-155. DOI: <https://doi.org/10.1016/j.ins.2024.120155>
10. Ma Z., Wang Y. (2019). Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons. *IEEE Transactions on Evolutionary Computation*, 23(6), 2019. PP. 972-986. DOI: <https://doi.org/10.1109/TEVC.2019.2896967>
11. Pyrih Y., Kaidan M., Tchaikovskiy I., Pleskanka M. (2019). Research of Genetic Algorithms for Increasing the Efficiency of Data Routing. *3rd International Conference on Advanced Information and Communications Technologies (AICT)*, Lviv, Ukraine, 2019. PP. 157-160. DOI: <https://doi.org/10.1109/AIACT.2019.8847814>
12. Qian H., Wu Y., Qin R., An X., Chen Y., Zhou A. (2025). Provable space discretization based evolutionary search for scalable multi-objective security games. *Swarm and Evolutionary Computation*, Vol 92, 101770, 2025. DOI: <https://doi.org/10.1016/j.swevo.2024.101770>
13. Seada H., Deb K. (2016). A unified evolutionary optimization procedure for single, multiple, and many objectives. *IEEE Transactions on Evolutionary Computation*, 20(3), 2016. PP. 358–369. DOI: <https://doi.org/10.1109/TEVC.2015.2459718>
14. Skarlat O., Nardelli M., Schulte S., Borkowski M., Leitner P. (2017). Optimized IoT service placement in the fog. Special Issue Paper. Open access. SOCA, Volume 11, 2017. PP. 427-443. DOI: <https://doi.org/10.1007/s11761-017-0219-8>
15. Takagi T., Takadama K., Sato H. (2020). Incremental lattice design of weight vector set. *2020 Genetic and Evolutionary Computation Conference Companion, GECCO '20*, New York, USA, Association for Computing Machinery, 2020. PP. 1486-1494. DOI: <https://doi.org/10.1145/3377929.3398082>
16. Vesikar Y., Deb K., Blank J. (2018) Reference point-based NSGA-III for preferred solutions. *In 2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018. PP. 1587-1594. DOI: <https://doi.org/10.1109/SSCI.2018.8628819>
17. Zitar R. (2022). A Review of the Genetic Algorithm and JAYA Algorithm Applications. *15-th International Congress on Image and Signal Processing, (CISP-BMEI)*, Beijing, China, 2022. PP. 1-7. DOI: <https://doi.org/10.1109/CISP-BMEI56279.2022.9980332>