



Международный журнал информационных технологий и энергоэффективности

Сайт журнала: <http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.7

КОМБИНИРОВАННЫЙ МЕТОД СТРУКТУРНОЙ ОПТИМИЗАЦИИ ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ

Петров А.С.

ФГБОУ ВО "МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н.Э. БАУМАНА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)", Москва, Россия, (105005, город Москва, 2-Я Бауманская ул, д. 5 стр. 1), e-mail: bauman@bmstu.ru

В статье рассматривается комбинированный генетический алгоритм применительно к структурной оптимизации сегмента локальной сети посредством максимизации пропускной способности и достоверности передаваемой информации (минимизации BER) при минимизации количества узлов сети с использованием различных методов многокритериальной оптимизации по Парето. Представлены результаты сравнения методов NSGA-II, VEGA, FFGA (MOGA), NPGA И SPEA-II при одних и тех же параметрах сети.

Ключевые слова: Многокритериальная оптимизация, комбинированный метод, генетический алгоритм с адаптивной мутацией, локальная сеть, NSGA-II, VEGA, FFGA (MOGA), NPGA, SPEA-II.

COMBINED METHOD OF STRUCTURAL OPTIMIZATION OF A LOCAL COMPUTER NETWORK

Petrov A.S.

BAUMAN MOSCOW STATE TECHNICAL UNIVERSITY (NATIONAL RESEARCH UNIVERSITY), Moscow, Russia, (105005, Moscow, 2nd Baumanskaya ul, 5 bld. 1), e-mail: bauman@bmstu.ru

The article discusses a combined genetic algorithm in relation to the structural optimization of a local network segment by maximizing the throughput and reliability of transmitted information (minimizing BER) while minimizing the number of network nodes using various methods of multicriteria Pareto optimization. The results of comparison of NSGA-II, VEGA, FFGA (MOGA), NPGA AND SPEA-II methods with the same network parameters are presented

Keywords: Multicriteria optimization, combined method, genetic algorithm with adaptive mutation, local network, NSGA-II, VEGA, FFGA (MOGA), NPGA, SPEA-II.

Введение

В настоящее время компьютерные сети применяются в множестве областей человеческой жизни: научной, технической, социальной, медицинской, коммерческой и т.д. [1]. Исходя из этого, вычислительные сети являются одним из наиболее перспективных направлений развития информационных технологий. Структурная оптимизация компьютерной сети, в свою очередь, позволит снизить затраты на ее управление, а также улучшить производительность и надежность. Оптимизацию столь сложной системы необходимо производить одновременно по нескольким критериям, поэтому традиционные методы оптимизации зачастую не справляются с этой задачей. Для нахождения Парето-оптимального

решения, удовлетворяющего сразу нескольким критериям, хорошо подходят так называемые эволюционные алгоритмы (ЭА). Одним из наиболее популярных ЭА, применяемых для решения сложных задач комбинаторики и оптимизации, является генетический алгоритм (ГА), впервые описанный Дж. Голландом [2]. Суть данного алгоритма состоит в том, что оптимизация основана на принципах естественного отбора, в которых из начальной популяции выделяют наиболее подходящих по функции приспособленности потомков. Поиск решений происходит с использованием таких генетических операторов, как: скрещивание (кроссинговер), мутация и селекция.

Пусть есть n переменных x_1, x_2, \dots, x_n и их k независимых целевых функций $f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_k(x_1, x_2, \dots, x_n)$ [3]. Тогда задачу многокритериальной оптимизации в общем случае можно сформулировать следующим образом [4]:

$$y = f(x) = (f(x_1, x_2, \dots, x_n)) \rightarrow \text{optimal}, \begin{cases} g_j(x) \leq 0, j = \overline{1, r}, \\ h_i(x) = 0, j = \overline{r+1, M}, \end{cases} \quad (1)$$

где $x = (x_1, x_2, \dots, x_n) \in X$ - вектор решений, $y = (f_1(x), f_2(x), \dots, f_k(x)) \in Y_f$ - вектор целевых функций. При этом X - пространство решений, Y_f - критериальное пространство (область значений критериев).

Стоит отметить, что приведенная постановка задачи не дает никаких указаний в случае, если невозможно добиться оптимальных значений для всех целевых функций одновременно. Также из данной формулировки многокритериальной оптимизации нельзя сделать вывод о том, как именно сравнивать различные полученные во время оптимизации результаты.

Существует несколько наиболее популярных методов решения этой проблемы [4-5].

- Метод линейной свертки.

$$F(x) = \sum_{i=1}^k w_i f_i(x), \quad (2)$$

где $F(x)$ - общий критерий оптимизации, w_i - весовые коэффициенты, отражающие относительную "важность" каждого критерия, а $f_i(x)$ - критерии оптимизации.

- Принцип доминирования по Парето.

Согласно принципу доминирования по Парето параметр x_1 *доминирует* над параметром x_2 , если он по всем критериям не хуже, чем x_2 , и хотя бы по одному --- лучше. В таком случае параметр x_1 называется *доминирующим*, а x_2 --- *доминируемым*. Если ни один из параметров x_1, x_2 не доминирует над другим, то они называются *недоминируемыми*. Множество Парето --- это множество, состоящее из недоминируемых объектов.

Формальное определение множества Парето в критериальном пространстве Y_f можно записать в следующем виде:

$$P(x) = x^* \in X \mid x \in X: y(x) \geq y(x^*), \quad (3)$$

для задачи максимизации.

$$P(x) = x^* \in X \mid x \in X: y(x) \leq y(x^*), \quad (4)$$

для задачи минимизации.

Структурная оптимизация будет реализована с помощью методов, использующих принцип Парето-доминирования.

При наличии нескольких целевых функций в стандартном ГА используются различные методы многокритериальной оптимизации, направленные на модификацию этапов назначения пригодности и селекции. Наиболее популярные из них перечислены далее: NSGA-II (модифицированный Non-dominated Sorting Genetic Algorithm), VEGA (Vector Evaluated Genetic Algorithm), FFGA (MOGA, Fonseca and Fleming's Multiobjective Genetic Algorithm), NPGA (Niche Pareto Genetic Algorithm) и SPEA2 (модифицированный Strength Pareto Evolutionary Algorithm).

Критерии сравнения многокритериальных методов

В задаче структурной оптимизации сегмента локальной вычислительной сети при выборе многокритериального алгоритма необходимо учитывать следующие критерии (при одинаковых параметрах ГА и одинаковой конфигурации сети).

1. Средние значения функции приспособленности индивидов на каждом поколении в процессе работы алгоритма. Для каждого поколения k , $k = 1, 2, \dots, K$.

$$AvgThroughput_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \left(\frac{\sum_{j=1}^{M_{i,k}} C_{j,i,k}}{M_{i,k}} \right), \quad (5)$$

$$AvgBER_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \left(\frac{\sum_{j=1}^{M_{i,k}} BER_{j,i,k}}{M_{i,k}} \right), \quad (6)$$

где:

- $AvgThroughput_k$ - средняя пропускная способность индивидов в k -м поколении;
- $AvgBER_k$ - средняя надежность (BER) индивидов в k -м поколении;
- $AvgVertices_k$ - среднее количество вершин индивидов в k -м поколении;
- K - суммарное количество поколений;
- N_k - количество индивидов в k -м поколении;
- $M_{i,k}$ - количество ребер i -го индивида в k -м поколении;
- $C_{i,j,k}$ - значение пропускной способности j -го ребра i -го индивида в k -м поколении;
- $V_{i,k}$ - количество вершин i -го индивида в k -м поколении.

2. Максимальные и минимальные значения функции приспособленности на каждом поколении в процессе работы алгоритма. Для каждого поколения k , $k = 1, 2, \dots, K$.

$$maxThroughput_k = \max_{i=1}^{N_k} \left(\frac{\sum_{j=1}^{M_{i,k}} C_{j,i,k}}{M_{i,k}} \right), \quad (8)$$

$$minThroughput_k = \min_{i=1}^{N_k} \left(\frac{\sum_{j=1}^{M_{i,k}} C_{j,i,k}}{M_{i,k}} \right), \quad (9)$$

$$maxBER_k = \max_{i=1}^{N_k} \left(\frac{\sum_{j=1}^{M_{i,k}} BER_{j,i,k}}{M_{i,k}} \right), \quad (10)$$

$$minBER_k = \min_{i=1}^{N_k} \left(\frac{\sum_{j=1}^{M_{i,k}} BER_{j,i,k}}{M_{i,k}} \right), \quad (11)$$

$$\max Vertices_k = \max(V_{1,k}, V_{2,k}, \dots, V_{N_k,k}), \quad (12)$$

$$\min Vertices_k = \min(V_{1,k}, V_{2,k}, \dots, V_{N_k,k}), \quad (13)$$

где:

- $\max Throughput_k$ и $\min Throughput_k$ - максимальная и минимальная пропускные способности индивидов в k -м поколении соответственно;
- $\max BER_k$ и $\min BER_k$ - максимальная и минимальная надежности (BER) индивидов в k -м поколении соответственно;
- $\max Vertices_k$ и $\min Vertices_k$ - максимальное и минимальное количество вершин индивидов в k -м поколении соответственно;
- K - суммарное количество поколений;
- N_k - количество индивидов в k -м поколении;
- $M_{i,k}$ - количество ребер i -го индивида в k -м поколении.
- $C_{i,j,k}$ - значение пропускной способности j -го ребра i -го индивида в k -м поколении;
- $V_{i,k}$ - количество вершин i -го индивида в k -м поколении.

3. Номер поколения, на котором получено конечное решение.

$$k_{target} = \min\{k \mid \exists i \in N_k : (C_{i,k} = C_{goal}) \wedge (BER_{i,k} = BER_{goal}) \wedge (V_{i,k} = V_{goal})\}, \quad (14)$$

где:

- k_{target} - номер поколения, на котором получено конечное решение;
- $C_{i,k}, BER_{i,k}, V_{i,k}$ - значения пропускной способности, надежности и количества вершин i -го индивида в k -м поколении соответственно;
- $C_{goal}, BER_{goal}, V_{goal}$ - значения пропускной способности, надежности и количества вершин конечного решения соответственно.

4. Качество полученного решения - значения функции приспособленности полученного в результате работы алгоритма решения.

5. Наличие застреваний в локальном оптимуме.

6. Способность поддержания разнообразия решений.

Выбор критерия 1 обусловлен следующими факторами:

- помогает отслеживать изменения популяции от поколения к поколению;
- позволяет оценить эффективность алгоритма в целом и его способность к улучшению решений;
- помогает идентифицировать моменты застревания в локальных оптимумах (минимальные изменения лучших и худших значений функции приспособленности индивидов в каждом поколении при постоянном среднем значении фитнес-функции может указывать на то, что индивиды начинают сходиться к одному и тому же решению [6]).

С помощью критерия 2 можно оценить пространство поиска решений по конкретному критерию в каждом из поколений, а с помощью критерия 3 можно сравнить скорости сходимости алгоритмов. Критерии 4-6 помогают проанализировать, насколько корректно отрабатывают операторы кроссовера и мутации в процессе работы генетических алгоритмов.

Классический вероятностный генетический алгоритм

Сложность генетического алгоритма состоит в использовании множества необходимых параметров, неправильная настройка которых может повлиять на его эффективность. Поскольку подбор корректных значений параметров является нетривиальной задачей, было создано множество модификаций алгоритма, упрощающих настройку ГА. Одна из таких модификаций - вероятностный генетический алгоритм (ВГА).

ВГА - это традиционный ГА, сформулированный в терминах теории псевдодвулевой оптимизации [7]. Модификация заключается в явном вычислении компонентов вектора вероятностей и замены оператора скрещивания на оператор случайного выбора в соответствии с построенным распределением. Такие изменения в работе ГА помогают упростить настройку параметра, отвечающего за вероятность кроссовера.

Работу базового ВГА можно описать следующими шагами.

- 1) Генерация случайным образом начальной популяции решений.
- 2) Выбор r наиболее пригодных индивидов текущей популяции (родителей).

Вычисление вектора вероятностей по формуле 15.

$$\bar{P} = (p_1, \dots, p_n), p_j = P\{x_j = 1\} = \frac{1}{r} \sum_{i=1}^r x_j^i, j = \overline{1, n}, \quad (15)$$

где n - длина хромосомы, x_j^i - j -й бит i -го индивида.

- 3) Формирование популяции потомков в соответствии с распределением \bar{P} .
- 4) Формирование новой рабочей популяции из популяции родителей и потомков.
- 5) Применение мутации к новой популяции.
- 6) Повторение пунктов 2-6 до выполнения условий остановки.

Ниже приведены основные преимущества вероятностного генетического алгоритма по сравнению с стандартным ГА:

- многокритериальная оптимизация;
- меньшее число настраиваемых параметров;
- в общем случае является более надежным и эффективным по трудоемкости;
- множество вариаций многокритериальных методов.

Однако ВГА также имеет несколько недостатков:

- более сложная реализация и вычисления из-за использования вероятностных операторов и стохастического выбора родителей;
- возможность получения менее точных решений.

Вероятностный генетический алгоритм с адаптивной мутацией

Данный алгоритм является модификацией ВГА посредством автоматического расчета вероятности мутации на каждом шаге, вследствие чего у пользователя отсутствует необходимость настраивать этот оператор вручную [4].

Вероятность мутации рассчитывается по формуле 16.

$$P_m = \frac{S}{n}, S = \begin{cases} \frac{n}{2}, (X_k = 0) \cup \left(\frac{X_{k-1}}{2X_k} \geq \frac{n}{2}\right) \\ \left(\frac{X_{k-1}}{2X_k}\right), \text{ иначе,} \end{cases} \quad (16)$$

где n - длина хромосомы, X_{k-i} - разброс точек в пространстве решений на $(k-i)$ -ом поколении, X_k - разброс точек в пространстве решений на k -ом поколении. Функция S определяет силу мутации.

Разброс точек в пространстве решений вычисляется по формуле 17.

$$X = \frac{1}{N} \sum_{n=1}^{\bar{N}} \frac{\Delta d^n}{M_n - m_n},$$

где:

- $\Delta d^n = \frac{1}{N'} \sum_{i,j} |d_{ij}^n - d^n|$ --- среднее отклонение расстояний между индивидами i и j от среднего расстояния d^n в пространстве решений по каждой из n переменных;
- $d_{ij}^n = |d_i^n - d_j^n|$ --- расстояние между индивидами i и j по каждой из n переменных в пространстве решений;
- i, j - номера индивидов в популяции;
- N' - число всех возможных пар индивидов;
- \bar{N} - количество переменных (альтернатив) в решаемой задаче;
- M_n и m_n - максимально и минимально возможные значения n -й переменной во всей допустимой области соответственно.

Одной из особенностей такой мутации является то, что ее вероятность пропорционально зависит от плотности группировки решений-индивидов около какого-либо локального оптимума.

Главным преимуществом данного алгоритма является адаптивная мутация, а главным минусом, в свою очередь, - сложность реализации.

Структурная оптимизация сегмента локальной вычислительной сети

Для структурной оптимизации сегмента ЛВС в данной статье будет использоваться ВГА с адаптивной мутацией, поскольку в процессе работы именно этого алгоритма параметр мутации настраивается автоматически. Локальная сеть в условиях поставленной задачи может быть формализована взвешенным неориентированным графом, каждая вершина которого соответствует узлу сети, а каждое ребро - каналу связи. Поскольку оптимизация проводится на основании пропускной способности (в МБ/с) и надежности (BER) каналов связи, эти две характеристики следует хранить на каждом из ребер графа.

Так как в вершинах графа не хранятся никаких характеристик, в процессе мутации можно случайным образом удалять одну из вершин индивида (или удалять ребра). На первый взгляд может показаться, что следует заведомо удалять ребра с самым большим значением пропускной способности и самым маленьким значением BER, однако такая мутация приведет к преждевременному схождению алгоритма к локальным оптимумам, что противоречит основной идее использования ГА. Особой разницы между случайным удалением вершин или ребер нет, так как независимо от этого во время вычисления фитнес-функции необходимо дополнительно проверять связность подграфа и штрафовать решения, ведущие к несвязным графам. Это позволит обеспечить связность итогового решения. В сегменте сети могут находиться как сетевые устройства (коммутаторы, маршрутизаторы и т. д.), так и конечные устройства (компьютеры, принтеры, сканеры, серверы и т. д.). Определенные типы узлов сети соединяются конкретными типами каналов связей, поэтому при оптимизации сети важно не переставлять каналы связи местами (возможно, что конкретное ребро не подходит для соединения других вершин графа). Алгоритм является стохастическим, поэтому во избежание получения различных результирующих графов сети следует закрепить случайное зерно генератора случайных чисел (*random.seed(42)*).

Для оптимизации сети и сравнения многокритериальных алгоритмов были использованы следующие параметры:

- размер начальной популяции - 100;
- количество поколений - 15;
- вероятность кроссовера (для каждого гена в хромосоме) - 0.5;
- начальная и конечная вероятности мутации (для каждого гена в хромосоме) - 0.01 и 0.3 соответственно.

Использование более высоких значений параметров может привести к тому, что в результате оптимизации граф (при заданной начальной конфигурации) будет иметь слишком маленькое количество узлов или каналов связи.

На Рисунках 1,2 представлены сегмент сети до структурной оптимизации и после.

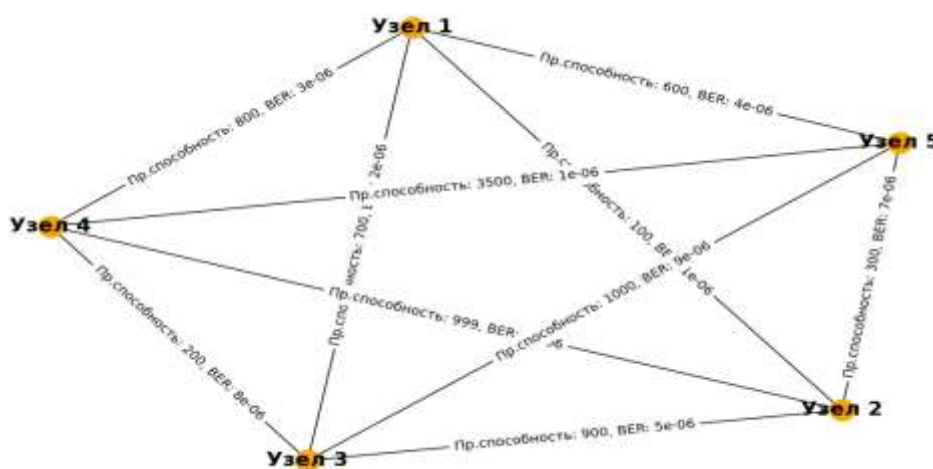


Рисунок 1 – Сегмент ЛВС до оптимизации

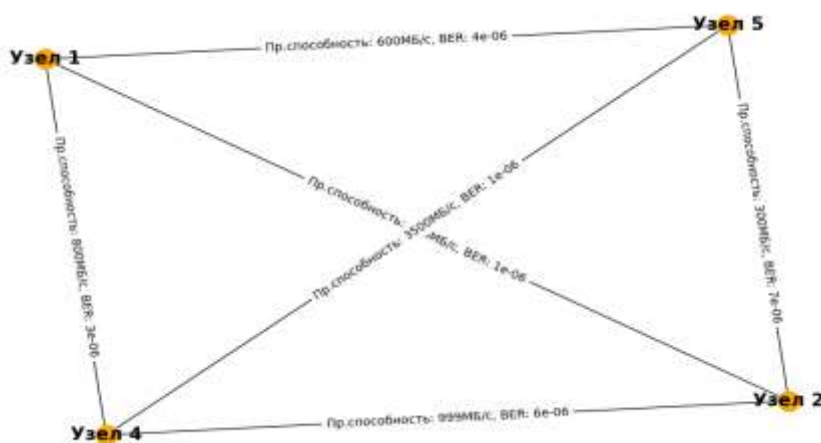


Рисунок 2 – Сегмент ЛВС после оптимизации

На Рисунках 3-8 приведены графики зависимости максимальных, средних и минимальных значений пропускной способности, BER и количества вершин популяции в зависимости от номера поколения. По этим графикам можно сделать выводы о критериях 2--6 для каждого из алгоритма.

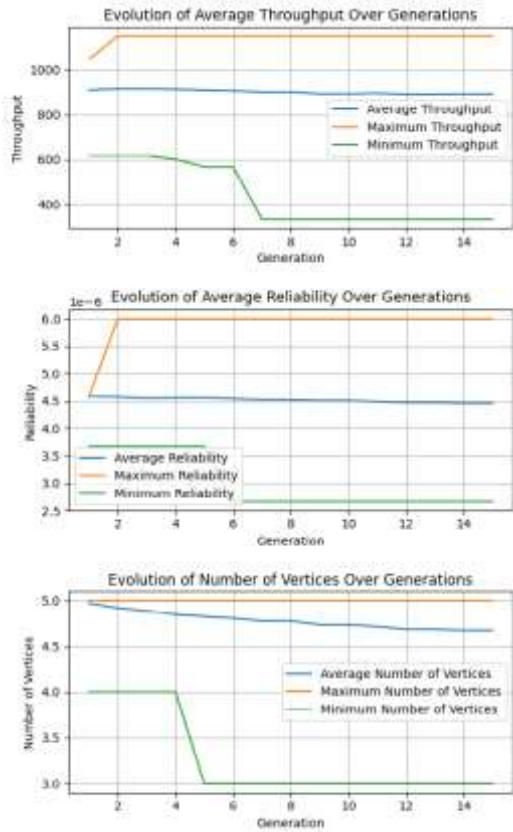


Рисунок 3 – Зависимость функции приспособленности от поколения в NSGA-II

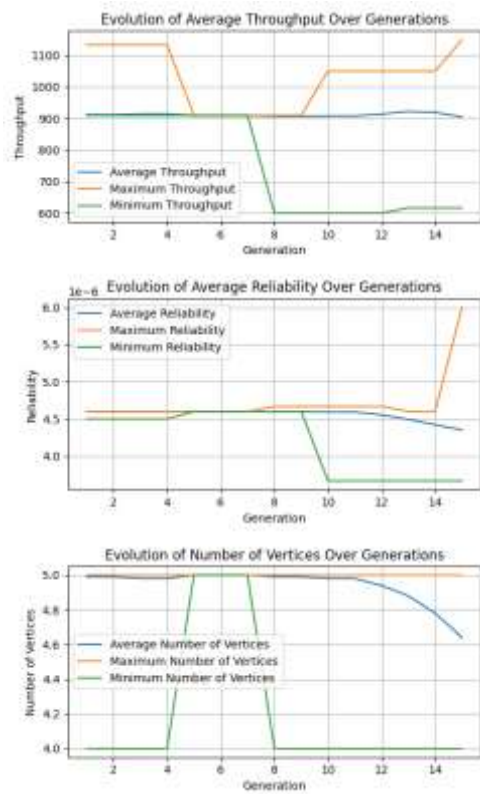


Рисунок 4 – Зависимость функции приспособленности от поколения в VEGA

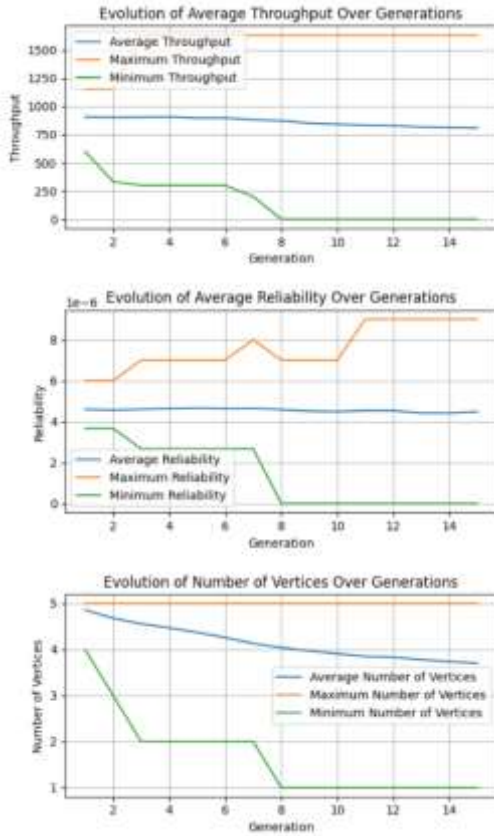


Рисунок 5 – Зависимость функции приспособленности от поколения в MOGA

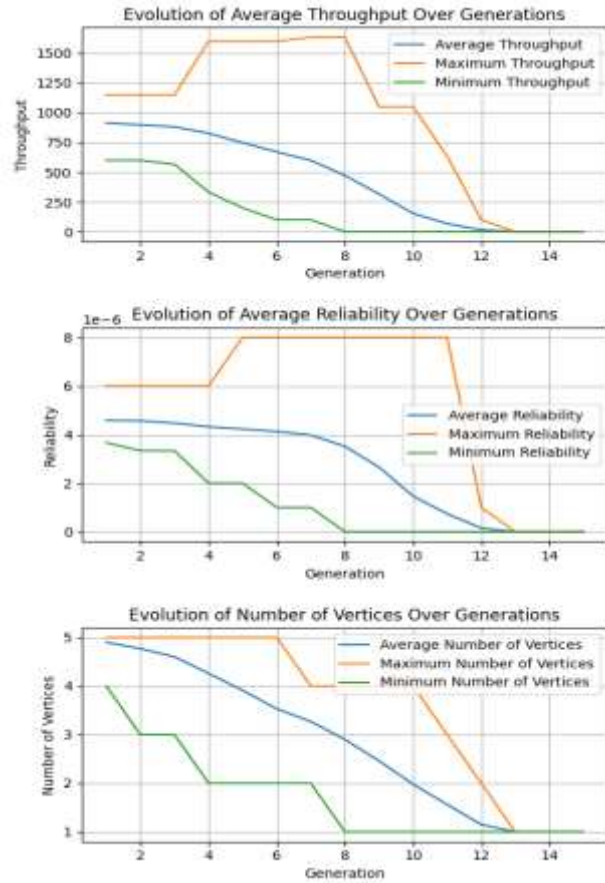


Рисунок 6 – Зависимость функции приспособленности от поколения в NPGA

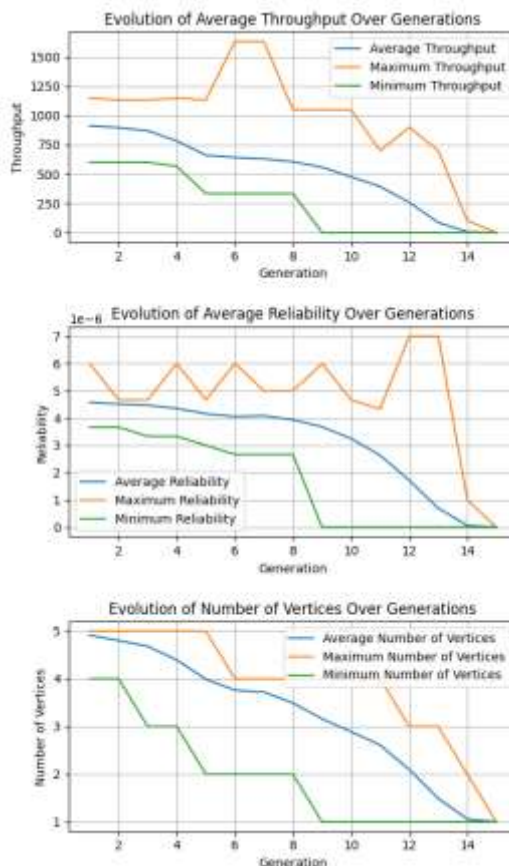


Рисунок 7 – Зависимость функции приспособленности от поколения в SPEA-II

По графикам видно, что наименее склонными к наличию застреваний в локальном оптимуме (критерии 1, 2, 5) при заданных параметрах являются MOGA и SPEA-II, однако учитывая малое разнообразие значений пропускной способности и надежности исходного графа можно сделать вывод о том, что алгоритм NSGA-II также не склонен к данной проблеме.

Разнообразие индивидов (критерий 6) присутствует в каждом из алгоритмов, поскольку ни в одной схеме нет ни единого критерия, среднее значение функции приспособленности которого оставалось бы постоянным. Стоит отметить, что даже малое изменение значений фитнес-функции при данных параметрах и конфигурации сети считаются значительными. В алгоритмах NPGA и SPEA-II индивиды более разнообразны, но это приводит к слишком большому уменьшению количества вершин при заданных параметрах.

Сравнить многокритериальные методы по критерию 4 исходя из полученных графиков не получится. Для этого требуется провести дополнительные замеры. Так как алгоритм обладает стохастической природой, для анализа эффективности многокритериальных схем каждый алгоритм независимо запускался по 100 раз. Затем, исходя из ста полученных исходов, рассчитывалась средняя величина функции приспособленности итоговых решений оптимизации. Во время работы различных алгоритмов значения функции приспособленности лучшего индивида на каждой итерации улучшались или оставались примерно постоянными с течением времени, что может свидетельствовать о сходимости алгоритма и, как следствие, его правильной работе [8]. Результаты замеров приведены в Таблице 1.

Таблица 1 – Значения функции приспособленности полученного в результате работы алгоритма решения

	Многокритериальная схема				
	NSGA-II	VEGA	FFGA (MOGA)	NPGA	SPEA-II
Пропускная способность	1098.563075 9453447	904.170029 7999148	975.825194 3072648	1010.825194 3072648	1088.688319 4139193
BER	4.0985065141 40451e-06	4.2561089825 45764e-06	4.0476251830 96482e-06	4.47391826 54093e-06	4.29384908424 90845e-06
Количество вершин	3.790943755 9580554	4.17624521 0727969	2.5829478 1639427	3.812357246 3582749	3.629450549 4505493
Поколение, на котором получено конечное решение	12	13	14	13	14

Из таблицы видно, что при выбранных значениях параметров и конфигурации сети схема NSGA-II показывает лучший результат по пропускной способности среди рассмотренных алгоритмов, в то время как VEGA показывает наименьшую пропускную способность. По критерию минимизации BER схема NSGA-II находится на втором месте, при этом количество вершин не сильно отличается от других методов. Несмотря на то, что алгоритмы находят оптимальные решения почти с одинаковой скоростью, NSGA-II опережает остальных на 1-2 поколения.

Исходя из вышеперечисленного, можно сделать следующий вывод: при структурной оптимизации сегмента локальной вычислительной сети вероятностным генетическим алгоритмом с адаптивной мутацией наиболее эффективным многокритериальным алгоритмом является вероятностный генетический алгоритм с адаптивной мутацией, использующий NSGA-II.

Заключение

В статье была проведена структурная оптимизация сегмента локальной вычислительной. Также было проведено сравнение многокритериальных методов на основании выделенных критериев. при структурной оптимизации сегмента локальной вычислительной сети вероятностным генетическим алгоритмом с адаптивной мутацией наиболее эффективным многокритериальным алгоритмом является вероятностный генетический алгоритм с адаптивной мутацией, использующий NSGA-II.

Список литературы

1. Букатов А.А., Гуда С.А.. Компьютерные сети: расширенный начальный курс. Учебник для вузов. -- СПб, 2020г.
2. Коваленко Н.Н., Семенкина О.Е. "Метод самоконфигурирования для настройки генетического алгоритма комбинаторной оптимизации". Актуальные проблемы авиации и космонавтики, vol. 2, no. 14, 2018, С. 49-51
3. Вычислительные технологии Том 23, № 5, 2018 Сравнение генетических алгоритмов MOGA и NSGA-II на задаче оптимизации формы рабочего колеса гидротурбины А. К. Гарагулова*, Д. О. Горбачева, Д. В. Чирков.
4. Сопов Е. А., Сопов, С. А. (2011). Вероятностный генетический алгоритм решения сложных задач многокритериальной оптимизации с адаптивной мутацией и прогнозом множества Парето. Вестник Самарского государственного аэрокосмического

университета им. академика С.П. Королёва (национального исследовательского университета), (6), С.273-282.

5. Зинченко А. С., Болквядзе И. Р., Внучков Ю. А. Применение метода линейной свертки критериев при оптимизации финансового обеспечения деятельности организации. Вестник университета, no. 1, 2017, С. 113-117.
6. Городилов А.Ю., Данилова Е.Ю. " ". Вестник Пермского университета. Серия: Математика. Механика. Информатика, no. 4 (31), 2015, С. 84-90.
7. P.V. Galushin, and E.S. Semenkin. «The asymptotic probabilistic genetic algorithm». Сибирский аэрокосмический журнал, no. 5, 2009, С. 45-49.
8. Alcaraz, M. Landete, J. F. Monge and J. L. Sainz-Pardo. Multi-objective evolutionary algorithms for a reliability location problem. Eur. J. Oper. Res., vol. 283, no. 1, pp. 83-93, May 2020.

References

1. Bukatov A.A., Guda S.A. Computer Networks: An Extended Introductory Course. Textbook for universities. -- St. Petersburg, 2020.
 2. Kovalenko N.N., Semenkin O.E. "SELF-CONFIGURATION METHOD FOR SETTING UP A GENETIC ALGORITHM OF COMBINATORIAL OPTIMIZATION". Current Problems in Aviation and Cosmonautics, vol. 2, no. 14, 2018, pp. 49-51.
 3. Garagulova A.K., Gorbacheva D.O., Chirkov D.V. "Comparison of MOGA and NSGA-II Genetic Algorithms in the Optimization Task of the Shape of a Hydraulic Turbine Runner Blade." Computational Technologies Volume 23, No. 5, 2018.
 4. Sopov E.A., Sopov S.A. (2011). "A Probabilistic Genetic Algorithm for Solving Complex Multi-Criteria Optimization Problems with Adaptive Mutation and Prediction of the Pareto Set". Bulletin of the Samara State Aerospace University named after academician S.P. Korolev (National Research University), (6), 273-282.
 5. Zinchenko A.S., Bolkvadze I.R., Vnuchkov Y.A. "Application of the Method of Linear Convolution of Criteria in the Optimization of Financial Support for the Activities of an Organization". University Bulletin, no. 1, 2017, pp. 113-117.
 6. Gorodilov A.Y., Danilova E.Y. " ". Bulletin of Perm University. Series: Mathematics. Mechanics. Informatics, no. 4 (31), 2015, pp. 84-90.
 7. Galushin P.V., Semenkin E.S. "The Asymptotic Probabilistic Genetic Algorithm". Siberian Aerospace Journal, no. 5, 2009, pp. 45-49.
 8. Alcaraz, M., Landete, J. F., Monge, J. L., Sainz-Pardo, J. L. "Multi-objective Evolutionary Algorithms for a Reliability Location Problem". European Journal of Operational Research, vol. 283, no. 1, pp. 83-93, May 2020.
-