

ISSN 2949-5598

ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ

Том 32 № 3 2025

Новосибирск
Издательство Института математики

ГИБРИДНЫЙ АЛГОРИТМ ДЛЯ ДВУХКРИТЕРИАЛЬНОЙ ЗАДАЧИ ОПТИМИЗАЦИИ ТРАФИКА В СЕТИ

А. Д. Юськов^{1, a}, И. Н. Кулаченко^{2, b},
А. А. Мельников^{2, c}, Ю. А. Кочетов^{2, d}

¹ Новосибирский гос. университет,
ул. Пирогова, 2, 630090 Новосибирск, Россия

² Институт математики им. С. Л. Соболева,
пр. Акад. Коптюга, 4, 630090 Новосибирск, Россия
E-mail: ^aa.yuskov@g.nsu.ru, ^bink@math.nsc.ru,
^cmelnikov@math.nsc.ru, ^djkochet@math.nsc.ru

Аннотация. Рассматривается задача оптимизации трафика в сети передачи данных. Для моделирования трафика используется имитационная модель. Пути передачи задаются неявно весами дуг. Если поток по дуге превышает её пропускную способность, то дуга считается перегруженной. Задача состоит в минимизации двух целевых функций: числа перегруженных дуг и расстояния от исходного вектора весов при соблюдении ограничений на суммарный поток в сети и появление новых перегруженных дуг. Предложена двухстадийная эволюционная схема, включающая алгоритм локального поиска по окрестностям большой мощности для получения стартового приближения границы Парето. Лучшее соседнее решение ищется при помощи оригинальной модели целочисленного линейного программирования. Проведено сравнение предложенного подхода с лучшими эволюционными алгоритмами на примерах с 628 каналами и 1324 запросами, и показано, что новая схема демонстрирует результаты, статистически лучшие на 15–49% по многим показателям качества (9 из 10). Табл. 3, ил. 6, библиогр. 42.

Ключевые слова: оптимизация «чёрного ящика», матэвристика, поиск с чередующимися окрестностями, OSPF, эволюционный алгоритм.

Введение

В связи с постоянно растущим объёмом интернет-трафика важно эффективно управлять сетевыми ресурсами, что может быть достигнуто

© А. Д. Юськов, И. Н. Кулаченко, А. А. Мельников, Ю. А. Кочетов, 2025

путём грамотной маршрутизации трафика по каналам. Для выбора путей маршрутизации трафика существуют различные протоколы. Методы управления трафиком включают в себя настройку весов каналов (например, в протоколах Open shortest path first (OSPF) и Intermediate system to intermediate system (IS-IS) [1]), использование многопротокольной коммутации по меткам (multiprotocol label switching, MPLS) [2], использование централизованных контроллеров таких, как программно определяемая сеть (software-defined networking, SDN) [3] и сегментная маршрутизация [4].

Такие протоколы, как MPLS, позволяют явно задавать пути для запросов. Это позволяет производить тонкую настройку сети и добиваться большой эффективности. Например, в работах [5, 6] обсуждаются различные точные методы и эвристики для построения маршрутов в сети. Однако такой подход требует больших вычислительных ресурсов для нахождения пути для каждого запроса и из-за этого плохо масштабируется на большие сети. Также в случае отказа соединения или узла придётся заново пересчитать маршруты для всех запросов, которые должны были пройти по этим каналам.

В данной работе рассматриваются протоколы, которые настраивают-ся путём задания весов сетевых соединений. Часто эти протоколы используют стратегию маршрутизации по кратчайшему пути [1]. Однако в работе не используются какие-либо предположения о структуре протокола. Это связано с тем, что протоколы, используемые на практике, может быть трудно явно записать математически, но потоки трафика, создаваемые протоколом, часто могут быть смоделированы с помощью компьютерной программы. Такие задачи, в которых мы не можем явно оценить целевую функцию и ограничения, называются задачами оптимизации «чёрного ящика» [7–9]. Способ маршрутизации, основанный на кратчайших путях, может быть не таким гибким, как например MPLS, так как не позволяет задавать различные маршруты для отдельных пакетов. Однако он более прост в настройке и эксплуатации, так как требует только настройки весов соединений, а также лучше масштабируется и более устойчив к отказам оборудования.

В протоколе OSPF основной задачей является настройка весов каналов. Это можно сделать, следуя некоторым простым правилам, например установив веса обратно пропорционально пропускной способности канала [10]. Однако такое решение может быть не оптимальным, поэтому существуют также метаэвристические подходы к решению данной задачи. Например, в статье [11] авторы разработали эвристику локального поиска, используя специальную нелинейную функцию стоимости. Следует заметить, что предложенная целевая функция обрела большую популярность в литературе и используется во многих источниках, приведённых

далее. В этой же работе авторы приводят некоторые оценки того, насколько маршрутизация, основанная на весах, может отличаться от оптимальной. В [12] представлен генетический алгоритм для этой задачи поиска весов. Авторы [13] разработали алгоритм реагирования на изменяющиеся условия в сети. Предложенный алгоритм пытается исправить неудовлетворительное состояние сети, возникшее из-за отказа каналов или изменения спроса, путём небольших изменений весов. В последние годы растущий интерес к машинному обучению побудил исследователей применить эти методы и к решению задач маршрутизации трафика [14].

Компании могут преследовать различные цели при построении и изменении сети. В большинстве случаев целью задач маршрутизации трафика является минимизация максимальной нагрузки на канал связи [15], уменьшение задержек в сети [16], улучшение балансировки нагрузки [17] или энергоэффективности [18]. Также популярна целевая функция в виде упомянутой ранее специальной нелинейной функции стоимости [11].

Из-за многокритериального характера задач маршрутизации существует множество статей, посвящённых оптимизации нескольких целевых функций, а наиболее популярным методом решения таких задач являются генетические алгоритмы, так как они естественным образом используют популяцию различных решений. Например, в [19] авторы применяют генетический алгоритм NSGA-II для оптимизации затрат и балансировки нагрузки. Авторы используют точный метод, чтобы получить решения для небольших примеров, и сравнивают эффективность эвристики с решениями на истинной границе Парето. В [20] обсуждается использование эволюционных алгоритмов SPEA2 и NSGA-II для минимизации перегрузки и задержки в сети. Авторы предлагают схему оптимизации, которая должна помочь сетевым администраторам выбрать подходящую конфигурацию для удовлетворения требований. В [18] оптимизируются балансировка нагрузки и энергоэффективность. Авторы предлагают эвристическую схему и сравнивают её с оптимальной производительностью сети. Сравнение алгоритмов SPEA2 и NSGA-II для задачи определения весов с возможными сбоями соединения представлено в [21]. Авторы обнаружили, что алгоритм NSGA-II демонстрирует лучшие общие результаты для больших задач. Более подробный обзор литературы о задаче маршрутизации трафика можно найти в [22].

Также можно отметить, что существует важная задача, являющаяся расширением данной: задача планирования сети, в которой требуется определить, какие соединения будут существовать между узлами и их пропускную способность. В этой задаче обычно учитывается как физическая топология сети, на которой находятся проложенные кабели, так и логическая топология, состоящая из сконфигурированных IP-каналов [23]. Обычно целью оптимизации является нахождение баланса

между стоимостью решения и его устойчивостью к отказам оборудования [24], хотя дополнительно существует большое количество различных критериев, которым должна удовлетворять сеть [25]. Для решения этой задачи также существуют различные методы: запись в виде ЦЛП-модели [26], генетические алгоритмы [27] и подходы с использованием машинного обучения [28]. Данная задача может представлять интерес для будущего исследования. Обзор различных направлений и задач можно найти в [29].

В данной работе рассматривается задача оптимизации, целью которой является минимизация общего числа перегруженных каналов в сети путём корректировки весов каналов. Канал называется перегруженным, если поток по нему превышает его пропускную способность. Потоки вычисляются во время моделирования сетевого протокола, который назначает пути для запросов на основе взвешенных длин путей. В задаче также предпочтительно не изменять веса значительно, поскольку это может привести к неожиданному поведению при практической реализации. Стало быть, вторая целевая функция задачи — минимизировать расстояние до исходного вектора весов. После оптимизации существующей сети качество обслуживания клиентов должно быть не хуже, чем оно было до оптимизации. Тем самым если канал не был перегружен до оптимизации, то в полученном после оптимизации решении он должен оставаться таковым. Кроме того, суммарная нагрузка на сетевые каналы не должна увеличиваться. Эти два условия являются строгими ограничениями в задаче. Такие целевые функции и ограничения зависят от значений потока трафика, полученных с помощью имитационной модели.

Основным результатом работы является следующее. Для данной задачи маршрутизации трафика в сети передачи данных предложена двухстадийная эволюционная схема. Она включает в себя алгоритм локального поиска по окрестностям большой мощности для получения стартового приближения границы Парето. Лучшее соседнее решение ищется при помощи оригинальной модели целочисленного линейного программирования. Проведено сравнение предложенного подхода с лучшими эволюционными алгоритмами, и показано, что новая схема демонстрирует результаты, которые статистически лучше на 15–49% по многим показателям качества (9 из 10). Данная статья является расширенным вариантом работы, представленной на конференции ОРТИМА 2023 [30].

В разд. 1 приведена математическая модель задачи. В разд. 2 представлен подход, основанный на модели линейного программирования. Вычислительные эксперименты обсуждаются в разд. 3. Они включают в себя описание различных показателей качества в п. 3.1, описание способа запуска эволюционных алгоритмов в п. 3.2, проверку правильности выбора соотношения бюджетов в двухэтапной схеме в п. 3.4, сравнение

приведённых подходов в п. 3.5 и, наконец, оценку дисперсии качества получаемых решений в п. 3.6. Краткие выводы, приведённые в заключении, завершают статью.

1. Постановка задачи

В работе используются следующие обозначения:

- A — это множество дуг в графе, соответствующих каналам связи в сети;
- $\mathbf{w}^0 = (w_a^0)_{a \in A}$ — начальный вектор весов каналов;
- c_a — пропускная способность канала $a \in A$.

Оптимизируемые переменные записаны в виде вектора $\mathbf{w} = (w_a)_{a \in A}$, содержащего веса каналов. После задания этих весов можно смоделировать поведение сети и вычислить следующие её характеристики:

- $l_a(\mathbf{w})$ — суммарная нагрузка дуги a ;
- $o_a(\mathbf{w}) = \begin{cases} 1, & \text{если дуга } a \in A \text{ перегружена,} \\ 0 & \text{иначе;} \end{cases}$
- $no_a(\mathbf{w}) = \begin{cases} 1, & \text{если } o_a(\mathbf{w}) = 1 \text{ и } o_a(\mathbf{w}^0) = 0, \\ 0 & \text{иначе.} \end{cases}$

По величинам потока в сети вычисляются следующие характеристики решения:

- общее число перегруженных каналов $O(\mathbf{w}) = \sum_{a \in A} o_a(\mathbf{w})$;
- расстояние $D(\mathbf{w}) = \|\mathbf{w} - \mathbf{w}^0\|_{\ell_1}$ между начальным и текущим векторами весов дуг;
- суммарная загрузка сети $L(\mathbf{w}) = \sum_{a \in A} l_a(\mathbf{w})$;
- число новых перегруженных каналов $NO(\mathbf{w}) = \sum_{a \in A} no_a(\mathbf{w})$.

Используя эти обозначения, задачу можно сформулировать как двухкритериальную задачу целочисленного программирования с использованием «чёрного ящика»:

$$O(\mathbf{w}) \rightarrow \min, \quad (1)$$

$$D(\mathbf{w}) \rightarrow \min, \quad (2)$$

$$L(\mathbf{w}) \leq L(\mathbf{w}^0), \quad (3)$$

$$NO(\mathbf{w}) = 0, \quad (4)$$

$$\mathbf{w} \in W. \quad (5)$$

Формулы (1)–(5) показывают, что цель состоит в том, чтобы найти векторы весов каналов, которые минимизируют число перегруженных каналов и имеют наименьшее расстояние от исходного вектора \mathbf{w}^0 , при

условии, что решения должны обеспечивать общую нагрузку на каналы, не превышающую первоначальной, и не должны вызывать новых перегрузок каналов.

Вектор весов $\mathbf{w} \in W$ называется *допустимым*, если он удовлетворяет условиям (3) и (4). Для двух допустимых весовых векторов $\mathbf{w}^1, \mathbf{w}^2 \in W$ говорим, что \mathbf{w}^1 *доминирует* \mathbf{w}^2 , если $O(\mathbf{w}^1) \leq O(\mathbf{w}^2)$, $D(\mathbf{w}^1) \leq D(\mathbf{w}^2)$ и по крайней мере одно из этих неравенств строгое. Задача состоит в том, чтобы найти Парето-множество решений, т. е. множество всех возможных решений без доминирования в рамках модели (1)–(5). Набор допустимых решений $S \subseteq W$ называется *приближением к множеству Парето* или *аппроксимирующим множеством*, если в S нет доминируемых решений. Далее в разд. 3.1 будут представлены показатели эффективности, которые характеризуют качество аппроксимирующего множества с разных точек зрения.

2. Эвристика, основанная на модели ЦЛП

Эвристический подход, который представлен в этом разделе, основан на модели целочисленного линейного программирования. Установлено, что классический локальный поиск, изменяющий вес одной дуги за раз, быстро останавливается в локальном оптимуме. Одновременное изменение нескольких весов помогает решить эту проблему, но размер такой окрестности экспоненциально увеличивается с ростом количества изменяемых весов, и её просмотр становится невозможным в рамках выделенного вычислительного бюджета. Для того чтобы обойти этот недостаток, предложено предсказывать изменения потока в сети, вызванные изменением нескольких весов, по изменениям, вызванным изменениями веса одной дуги. Представленная далее математическая модель целочисленного линейного программирования способна предсказать такие изменения и выбрать лучшее решение в окрестности. Она использует значения изменений потока трафика в зависимости от изменений веса какого-либо канала и пытается подобрать комбинацию изменений нескольких весов одновременно так, чтобы минимизировать число перегруженных дуг. Так как изменения весов влияют друг на друга, решения, предсказанные моделью, могут отличаться по качеству от настоящих, однако эксперименты показывают, что предложенный подход позволяет находить хорошие решения.

2.1. Модель минимизации перегрузки. Путём изменения отдельного веса в векторе \mathbf{w} можно вычислить, как изменение веса одной дуги $e \in E \subseteq A$ влияет на нагрузку каждой дуги. Далее рассматриваем только подмножество всех весов, поскольку моделирование требует больших

вычислительных затрат, при этом желательно свести к минимуму число вызовов имитационной модели. Структура множества E будет представлена ниже в разд. 2.2. Пусть h обозначает размер шага, а \mathbf{w}^{e+} — весовой вектор, полученный из \mathbf{w} путём увеличения его e -й компоненты по формуле $w_e^{e+} = \min\{w_e^{\max}, w_e + h\}$. Аналогично e -я компонента вектора уменьшенного веса \mathbf{w}^{e-} равна $w_e^{e-} = \max\{w_e^{\min}, w_e - h\}$. Изменение нагрузки для каждой дуги $a \in A$ может быть вычислено в рамках одного запуска имитационной модели следующим образом:

$$l_a^{e+} = l_a(\mathbf{w}^{e+}) - l_a(\mathbf{w}), \quad l_a^{e-} = l_a(\mathbf{w}^{e-}) - l_a(\mathbf{w}).$$

Чтобы сформулировать модель ЦЛП для минимизации перегрузки, введём булевы переменные $(x_a)_{a \in A}$, указывающие, перегружена соответствующая дуга или нет, и булевы переменные $(\lambda_e^+), (\lambda_e^-)$, указывающие, увеличивается или уменьшается вес соответствующего канала.

С этими обозначениями модель минимизации перегрузки записывается следующим образом:

$$\sum_{a \in A} x_a \rightarrow \min, \quad (6)$$

$$\sum_{e \in E} (\lambda_e^+ + \lambda_e^-) \leq k, \quad (7)$$

$$\lambda_e^+ + \lambda_e^- \leq 1, \quad e \in E, \quad (8)$$

$$l_a(\mathbf{w}) + \sum_{e \in E} l_a^{e+} \lambda_e^+ + \sum_{e \in E} l_a^{e-} \lambda_e^- \leq c_a + o_a(\mathbf{w}^0) M x_a, \quad a \in A, \quad (9)$$

$$\sum_{a \in A} \left(l_a(\mathbf{w}) + \sum_{e \in E} l_a^{e+} \lambda_e^+ + \sum_{e \in E} l_a^{e-} \lambda_e^- \right) \leq L(\mathbf{w}^0), \quad (10)$$

$$\lambda_e^+, \lambda_e^-, x_a \in \{0, 1\}, \quad a \in A, e \in E. \quad (11)$$

Целевая функция (6) минимизирует число перегруженных каналов. Условия (7) ограничивают количество модификаций текущего весового вектора \mathbf{w} . Это необходимо, поскольку изменения компонентов вектора веса влияют друг на друга, и одновременное изменение нескольких компонентов приводит к непредсказуемым изменениям нагрузки. Поскольку изменения нагрузки, вызванные увеличением и уменьшением веса соединения, не противоположны друг другу, в оптимальное решение модели теоретически могут входить обе модификации одной дуги. Чтобы это предотвратить, ограничения (8) запрещают такое поведение. Ограничения (9) указывают, что соединения, которые не были перегружены изначально, не должны стать перегруженными после изменения веса. Эти ограничения, включающие достаточно большую константу M , гарантируют, что переменной x_a будет присвоено значение, равное единице, если

пропускная способность соответствующего канала a будет превышена. Наконец, неравенство (10) гарантирует, что общая нагрузка после изменения весов не превысит начального значения общей нагрузки $L(\mathbf{w}^0)$.

2.2. Поиск с чередующимися окрестностями. Здесь описан алгоритм, который использует модель, представленную в п. 2.1. Он напоминает схему спуска с чередующимися окрестностями с изменяемым шагом [31]. При такой аналогии окрестность решения \mathbf{w} включает решения, которые могут быть достигнуты путём изменения не более чем k элементов вектора \mathbf{w} на величину h . Лучшее решение из этой окрестности в предположении, что модификации различных дуг не влияют друг на друга, можно найти, взяв оптимальное решение для модели (6)–(11). Различные значения k и h задают различные окрестности. Введём множества K и H для возможных значений k и h . Пусть множество K зависит от двух параметров: $k_{\max} \in \mathbb{N}$ и категориального параметра $\text{RT} \in \{\text{fixed}, \text{decremental}, \text{exp}\}$, обозначающего тип диапазона:

$$K(\text{RT}) = \begin{cases} \{k_{\max}\}, & \text{если } \text{RT} = \text{fixed}, \\ \{k_{\max}, k_{\max} - 1, \dots, 1\}, & \text{если } \text{RT} = \text{decremental}, \\ \{k_{\max}, \lfloor k_{\max}/2^1 \rfloor, \lfloor k_{\max}/2^2 \rfloor, \dots, 1\}, & \text{если } \text{RT} = \text{exp}. \end{cases}$$

Пусть множество H зависит от параметров h_{\min} и h_{\max} следующим образом:

$$H(h_{\min}, h_{\max}) = \{2^i h_{\min} \mid i \in 0, 1, \dots, \lfloor \log_2 (h_{\max}/h_{\min}) \rfloor\}.$$

Для определения размера шага h алгоритм делит все дуги на группы в зависимости от начальных весов:

$$E_g = \{i \in A \mid 10^g \leq w_i^0 < 10^{g+1}\}, \quad G = \{E_g \mid g = 0, \dots, \lfloor \log_{10} \max_{i \in A} w_i^0 \rfloor\}.$$

В каждой группе используются границы $h_{\min,g} = \lceil \min \{w_i^0 \mid i \in E_g\} / 4 \rceil$ и $h_{\max,g} = 64h_{\min,g}$ для шага h . В экспериментах также рассматривается вариант без разделения на группы и $|G| = 1$. В этом случае устанавливаются параметры $h_{\min} = 1$ и $h_{\max} = 0,5 \max_{i \in A} w_i^0$. Схема поиска с чередующимися окрестностями на основе модели VNMS представлена алгоритмом 1.

Процедура локального улучшения `solveModel(\mathbf{w}, h, E, k)` заключается в решении модели (6)–(11) с соответствующими значениями \mathbf{w} , h , E и k . Алгоритм начинает с $g = 1$, $h = h_{\min,1}$, $k = k_{\max}$. Если новое решение, полученное после решения модели, не лучше старого, то параметры обновляются следующим образом. Сначала уменьшается значение k . Если оно становится меньше k_{\min} , то k сбрасывается до k_{\max} , а значение h удваивается. Если h оказывается больше, чем h_{\max} , то его значение устанавливается равным h_{\min} , и алгоритм переходит к следующей группе.

Алгоритм 1. Поиск с чередующимися окрестностями на основе модели

```

1: function VNMS( $G, H, K$ )
2:   for  $g = 1, \dots, |G|$  do
3:      $H \leftarrow H(h_{\min,g}, h_{\max,g})$ ;
4:     for all  $h \in H$  do
5:        $i \leftarrow 0$ ;
6:       while  $i \leq |K|$  do
7:          $w' \leftarrow \text{solveModel}(w, h, E_{g,p}, K[i])$ ;
8:         if  $w'$  недопустимое then  $\text{greedyFix}(w')$ ;
9:         if  $w'$  лучше, чем  $w$  then  $w \leftarrow w'$ ;
10:        else  $i \leftarrow i + 1$ ;
11:   return  $w$ ;

```

После завершения работы с последней группой алгоритм останавливается. Однако в ходе экспериментов он перезапускался с начала, пока бюджет вычислений оставался неисчерпанным.

Дополнительно в схеме реализован механизм быстрой починки решения, который задействуется, если решение на выходе модели ЦЛП недопустимо из-за перегрузки новых дуг. Указанный механизм жадно увеличивает веса вновь перегруженных дуг, при этом выполняется не более 200 итераций увеличения; эта процедура названа *greedyFix*. Также, чтобы ускорить поиск и внести разнообразие в алгоритм, оценивается только случайное подмножество соседей $E_{g,p}$, в которое каждое изменение попадает с вероятностью $p \in (0, 1]$.

Хотя такой эвристический подход может применяться сам по себе, большой интерес представляет гибридизация этой схемы с хорошо известными многокритериальными эволюционными алгоритмами (multi-objective evolutionary algorithm, MOEA). Оба метода имеют свои преимущества и недостатки. Эволюционные алгоритмы доказали свою способность находить решения, которые очень близки к множеству Парето. Однако им может быть трудно удовлетворить ограничения, в то время как подход на основе представленной модели, учитывает эти ограничения явным образом. Эвристический подход позволяет получить допустимые точки, далёкие от начального решения, однако он возвращает лишь небольшое число решений, причём часть из них могут находиться значительном расстоянии от истинной границы Парето. Следовательно, двухэтапная схема, в которой эволюционные алгоритмы используются для последующей оптимизации аппроксимации множества Парето, найденного с помощью подхода, основанного на модели, должна обеспечить хорошие результаты.

3. Вычислительные эксперименты

Все эксперименты, описанные в этом разделе, проводились на компьютере, оснащённом процессором Intel Core i7-8700 3,20 ГГц и 32 ГБ оперативной памяти, под управлением операционной системы Microsoft Windows 10 Pro. Для реализации модели, описанной в разд. 2, применены библиотека PuLP¹⁾ на языке Python и пакет CBC [32]. CBC был запущен в одном потоке, в то время как все остальные операции выполнялись параллельно с использованием всех доступных ядер.

Для оценки качества работы алгоритма было сгенерировано 13 тестовых примеров из предоставленного нам реального. Исходный пример содержит $|A| = 628$ каналов и 1324 запроса (пары источник-назначение). Сгенерированные примеры имеют ту же структуру графа и матрицу источник — назначение, что и исходный пример. Отличие заключается в векторе начальных весов \mathbf{w}^0 . В работе использованы различные схемы выбора начальных весов: выбор согласно рекомендации Cisco [10], нормализация весов исходного примера, перемешивание исходных весов и случайное равномерное распределение весов. В итоге максимальный вес в примерах варьируется от 10 до 10 000. Подробнее механизм генерации примеров описан в [30].

3.1. Показатели качества. Для численного анализа эффективности алгоритмов применяются показатели эффективности, разработанные специально для задач многокритериальной оптимизации [33] для сравнения приближений множеств Парето. Для описания показателей потребуются обозначения S или S_k , $k \in \mathbb{N}$, для приближений границы Парето, полученных с помощью исследуемых алгоритмов. В некоторых определениях показателей используется специальное множество решений R , называемое *эталонным множеством*, которое является точной Парето-границей или же заведомо достаточно хорошим приближением к ней. Вектор целевых функций для удобства можно обозначить через $F = (f_i)_{i \in I}$. В нашем случае $I = \{1, 2\}$, $f_1 = O$ и $f_2 = D$. Рассматриваются следующие показатели.

- *Гиперобъём* (hypervolume) определяется как объём в пространстве целевых функций, который доминируется приближением к границе Парето и ограничен сверху некоторой точкой. В качестве такой точки используем $(O(\mathbf{w}^0), \max_{\mathbf{w} \in R} D(\mathbf{w}))$.
- *Вклад* (contribution) — доля точек из эталонного множества R , которые присутствуют в S .

¹⁾ <https://github.com/coin-or/pulp>

- *Расстояние до эталона* (generational distance, GD) — это расстояние от S до R :

$$GD(S, R) = \frac{1}{|S|} \left(\sum_{\mathbf{w}^s \in S} \min_{\mathbf{w}^r \in R} \|F(\mathbf{w}^s) - F(\mathbf{w}^r)\|^p \right)^{\frac{1}{p}}.$$

- *Расстояние от эталона* (inverted generational distance, IGD) — расстояние от R до S : $IGD(S, R) = GD(R, S)$. Для GD и IGD используем $p = 2$.

- ε -Индикатор — это значение, необходимое для того, чтобы S аддитивно ε -доминировало R . Говорим, что вектор \mathbf{w}^1 аддитивно ε -доминирует \mathbf{w}^2 , если $f_i(\mathbf{w}^1) \leq \varepsilon + f_i(\mathbf{w}^2)$ для любого $i \in I$.

- *Максимальная ошибка* (maximum Pareto front error, MPFE) — максимальное расстояние точки из S до R :

$$MPFE(S, R) = \max_{\mathbf{w}^s \in S} \min_{\mathbf{w}^r \in R} \|F(\mathbf{w}^s) - F(\mathbf{w}^r)\|.$$

- *Показатели* R_1 и R_2 . Пусть S_1 и S_2 — две аппроксимации множества Парето, U — набор функций полезности $u: \mathbb{R}^m \rightarrow \mathbb{R}$. Для каждого $u \in U$ и $s = 1, 2$ пусть задано $u^*(S_s) = \min_{\mathbf{w} \in S_s} u(F(\mathbf{w}))$. Эти два показателя измеряют, в какой степени S_1 лучше, чем S_2 , по набору функций полезности U :

$$C(S_1, S_2, u) = \begin{cases} 1, & \text{если } u^*(S_1) < u^*(S_2), \\ 1/2, & \text{если } u^*(S_1) = u^*(S_2), \\ 0, & \text{если } u^*(S_1) > u^*(S_2); \end{cases}$$

$$R_1(S_1, S_2, U) = \frac{1}{|U|} \sum_{u \in U} C(S_1, S_2, u);$$

$$R_2(S_1, S_2, U) = \frac{1}{|U|} \sum_{u \in U} (u^*(S_1) - u^*(S_2)).$$

Если $R_1(S_1, S_2, U) > 0.5$, то считается, что S_1 лучше, чем S_2 . Аналогично $R_2(S_1, S_2, U) < 0$ соответствует тому, что S_1 показывает результаты лучше, чем S_2 .

- *Расстояние между решениями* (spacing) рассчитывается как

$$SP(S) = \sqrt{\frac{1}{|S|-1} \sum_{i=1}^{|S|} (\bar{d} - d_i)^2},$$

где $d_i = \min_{\mathbf{w}^j \in S \setminus \mathbf{w}^i} \|F(\mathbf{w}^i) - F(\mathbf{w}^j)\|_{\ell_1}$ — расстояние между точкой $\mathbf{w}^i \in S$ и ближайшей точкой аппроксимации множества Парето, полученной с помощью того же алгоритма, а \bar{d} — среднее значение d_i .

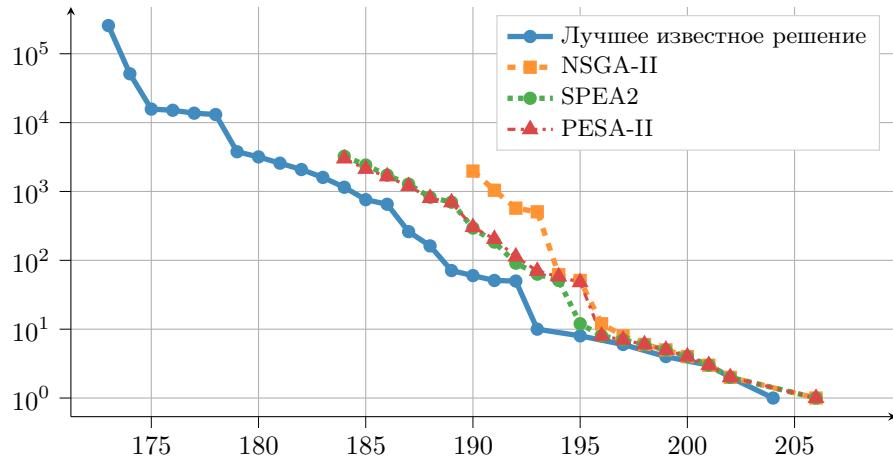


Рис. 1. Качество полученной границы Парето: без начального решения w^1

- Минимальное число перегруженных дуг равно $O_{\min}(S) = \min_{\mathbf{w} \in S} O(\mathbf{w})$.
- Число решений (cardinality) равно $|S|$.

3.2. Многокритериальные популяционные алгоритмы. В литературе одними из самых популярных для многокритериальной оптимизации являются алгоритмы, основанные на популяции решений [34], поэтому в этой работе также рассмотрены несколько наиболее широко

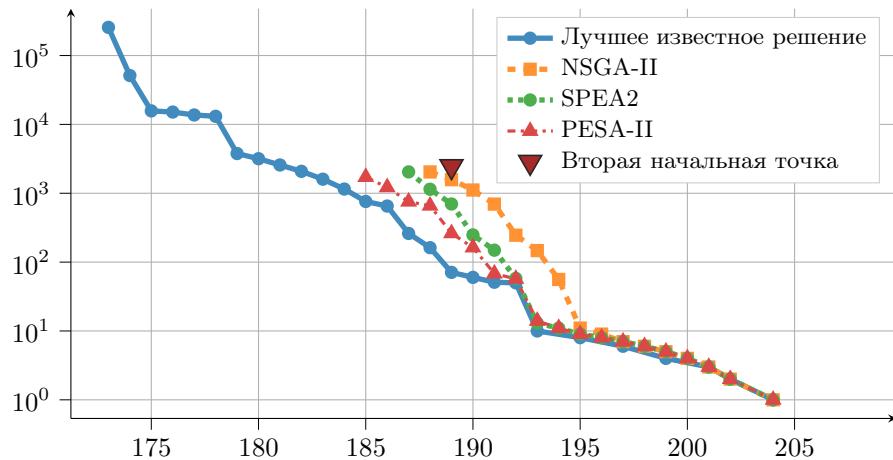


Рис. 2. Качество полученной границы Парето: среднее $O_{\min}(\mathbf{w}^1)$, среднее $D(\mathbf{w}^1)$

используемых многокритериальных эволюционных алгоритмов. А именно, применяются следующие многокритериальные алгоритмы из Java-библиотеки MOEA framework [35]:

- NSGA-II – nondominated sorting genetic algorithm II [36];
- SPEA2 – strength Pareto evolutionary algorithm 2 [37];
- PESA-II – Pareto envelope region-based selection algorithm [38];
- PAES – Pareto archived evolutionary strategy [39].

Также протестированы другие эволюционные алгоритмы, но они показали результаты хуже, чем перечисленные выше.

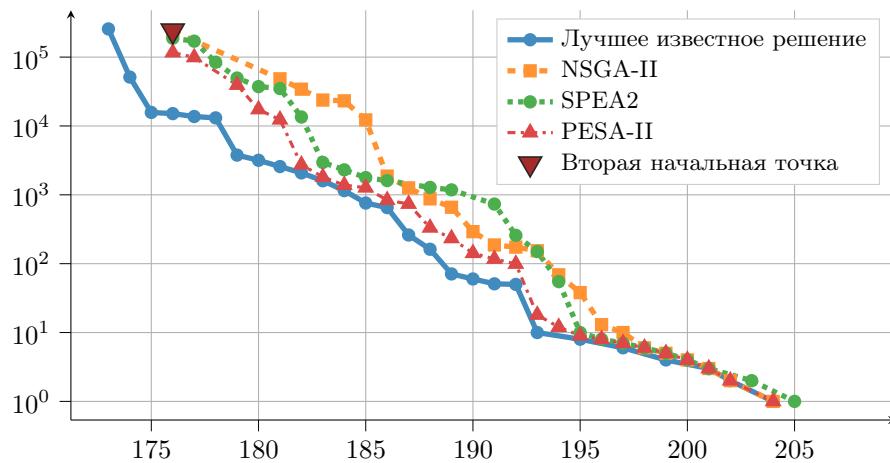


Рис. 3. Качество полученной границы Парето:
хорошее $O_{\min}(\mathbf{w}^1)$, среднее $D(\mathbf{w}^1)$

Стоит отметить, что алгоритмы, основанные на популяции, работают намного лучше, если им предоставляются несколько начальных решений $\{\mathbf{w}^0, \mathbf{w}^1, \dots, \mathbf{w}^m\}$, а не только \mathbf{w}^0 . На рис. 1–5 представлено поведение алгоритмов при двух начальных решениях $\{\mathbf{w}^0, \mathbf{w}^1\}$ для различных значений \mathbf{w}^1 , при этом по горизонтальной оси отложено число перегруженных каналов, а по вертикальной — расстояние до границы Парето. Все результаты соответствуют исходному примеру.

Как видно из рис. 1–5, качество результатов, достигаемых с помощью алгоритмов, сильно зависит от исходных решений. Желательно, чтобы в начальной популяции было по возможности близкое к \mathbf{w}^0 решение, которое имеет как можно меньшее значение $O(\mathbf{w})$. Последнее необходимо для того, чтобы иметь возможность находить промежуточные потенциальные решения.

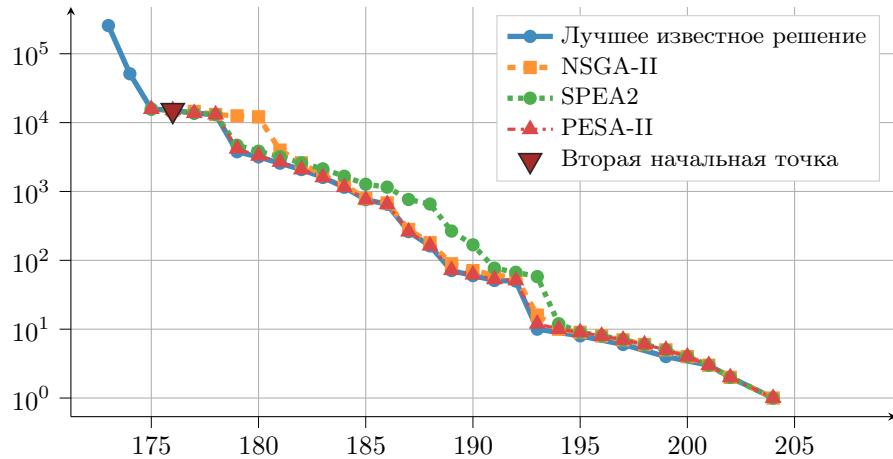


Рис. 4. Качество полученной границы Парето:
хорошее $O_{\min}(\mathbf{w}^1)$, хорошее $D(\mathbf{w}^1)$

Из-за присутствия ограничений в задаче эволюционным алгоритмам трудно исследовать пространство решений и находить хорошие отдалённые. Для повышения качества итоговых решения можно инициализировать популяцию с помощью решений, полученных при помощи метода, учитывающего эти ограничения, подобного тому, который представлен в разд. 2.

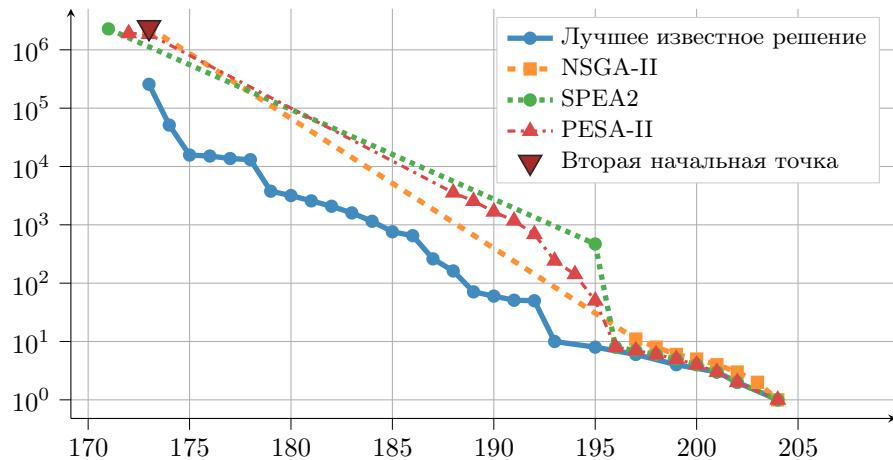


Рис. 5. Качество полученной границы Парето:
хорошее $O_{\min}(\mathbf{w}^1)$, плохое $D(\mathbf{w}^1)$

В предыдущей работе [30] было показано, что алгоритм PAES наилучшим образом среди эволюционных алгоритмов подходит для нахождения решений с малым числом перегруженных дуг. Там же было показано, что гибридизация других популяционных алгоритмов с PAES улучшает качество найденных решений, а среди таких вариантов выигрывает схема, в которой на 50 000 оцениваний целевой функции запускается PAES, после чего запускается алгоритм PESA-II с бюджетом 70 000 оцениваний целевой функции.

3.3. Выбор параметров. Чтобы определить хорошие значения параметров схемы VNMS, было решено использовать инструмент оптимизации гиперпараметров SMAC3 [42]. Так как схема может использоваться в двух вариантах — как самостоятельный алгоритм и в связке с PESA-II, были найдены два набора параметров: $RT = \text{exp}$, $k_{\max} = 8$, $p = 0,33$ с разделением на группы для первого варианта и $RT = \text{decremental}$, $k_{\max} = 4$, $p = 0,5$ без использования групп для второго. Схемы с этими параметрами будем называть $VNMS_{\text{fast}}$ и $VNMS_{\text{long}}$ соответственно.

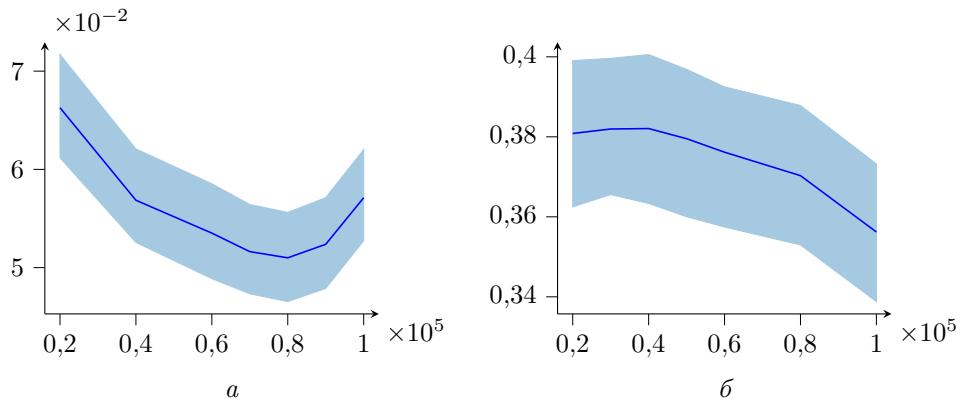


Рис. 6. Зависимость показателей качества от количества оцениваний целевой функции, выделенных на эволюционный алгоритм:
 а — расстояние от эталона, б — гиперобъём

3.4. Распределение бюджета между этапами. В этом эксперименте производится проверка, что соотношение вычислительных бюджетов, выделенных на построение решений с помощью схемы VNMS и на их последующее улучшение с помощью эволюционного алгоритма, выбрано верно. Для этого выполнена серия запусков с одинаковым суммарным бюджетом, равным, как и ранее, 120 000, но с разным соотношением бюджетов между этапами. На рис. 6 отражены зависимости среднего расстояния от эталона и среднего гиперобъёма от количества оцениваний

целевой функции, выделенных на эволюционный алгоритм. Остальной бюджет потрачен предложенной схемой на генерацию начальных решений. Закрашенная полоса обозначает доверительный интервал для уровня доверия 0,95.

Из графиков видно, что минимум расстояния, как и максимум объёма, достигается при бюджете, выделенном на эволюционный алгоритм, равном 80 000. Это говорит о том, что соотношение бюджетов, использованное ранее, близко к оптимальному, хотя можно добиться небольшого улучшения, если увеличить бюджет на улучшение решений эволюционным алгоритмом.

3.5. Сравнение схем. В этом пункте сравниваются схемы, выбранные ранее, а именно: PAES+PESA-II, VNMS_{fast}+PESA-II и VNMS_{long}. Первые две схемы двухэтапные, где часть бюджета выделяется на построение начальных решений, а оставшийся бюджет расходуется PESA-II, использующим эти решения в качестве стартовых. Для краткости обозначений назовём двухэтапные схемы PPESA и VPESA соответственно, а для алгоритма VNMS_{long} будем также использовать просто обозначение VNMS. Общий объём вычислений для всех трёх схем примерно одинаков и составляет 12 минут или 120 000 вычислений целевой функции.

Каждый алгоритм запущен по 10 раз, а затем с использованием результатов этих запусков применены два U-критерия Манна — Уитни [40] для каждого примера, каждой пары вариантов запусков и всех индикаторов, за исключением индикаторов R_1 и R_2 . Нулевая гипотеза состоит в том, что вероятность того, что значение показателя для первого алгоритма в попарном сравнении будет лучше значения для второго алгоритма, не превышает вероятности того, что значение показателя для

Таблица 1

Число побед в попарных сравнениях выбранных схем

Показатель	PPESA:VPESA	VPESA:VNMS	VNMS:PPESA
Гиперобъём	1:11	11:0	6:6
Вклад	1:7	11:0	1:8
Расст. до эталона	3:7	12:0	1:11
Расст. от эталона	1:12	11:0	9:3
ε -индикатор	0:11	2:2	0:12
Макс. ошибка	5:4	9:0	1:7
R_1	0:11	2:2	10:0
R_2	0:11	2:2	11:0
O_{\min}	0:12	0:3	13:0
Число решений	0:10	13:0	2:10

второго варианта будет лучше значения для первого варианта. Для индикаторов R_1 и R_2 проверены критерии знаковых рангов Уилкоксона [41] с нулевой гипотезой о том, что один вариант не лучше другого с точки зрения значений индикаторов R_1, R_2 (п. 3.1).

Говорим, что первый алгоритм выигрывает на данном примере по данному показателю, если нулевая гипотеза о том, что второй алгоритм не лучше первого, отвергается и принимается альтернативная гипотеза о том, что он лучше. Аналогично говорим, что первый алгоритм проигрывает, если принимается альтернативная гипотеза о том, что второй алгоритм лучше в этом случае и по данному показателю. В случае, если не принимается ни одна из альтернативных гипотез, говорим, что эти варианты эквивалентны. Результаты сравнений представлены в табл. 1. Первое число пары, разделённое двоеточием — это число побед первого алгоритма в паре, а второе число — число побед второго.

Как видно из табл. 1, двухэтапная схема с использованием VNMS_{fast} для генерации исходных решений работает наилучшим образом из проверенных вариантов. Схема VNMS_{long} также даёт хорошие результаты. Хотя гибридная схема, использующая эволюционные алгоритмы, аппроксимирует границу Парето вблизи исходного решения намного лучше, чем VNMS_{long}, последняя значительно превосходит эволюционные методы (включая PAES) в поиске решений с малыми значениями $O(\mathbf{w})$. Это приводит к улучшению расстояния от эталона, ε -индикатора и O_{\min} и сопоставимым результатам с точки зрения гиперобъёма. Схема VPESA использует преимущества обоих подходов, что приводит к получению более качественных решений.

Таблица 2

**Средние значения показателей
и относительное улучшение алгоритмов**

Показатель	PPESA	VNMS	VPESA	Улучш. VNMS	Улучш. VPESA
Гиперобъём	0,336	0,345	0,381	0,043	0,153
Вклад	0,164	0,113	0,211	-0,133	0,392
Расст. до эталона	0,010	0,020	0,008	-0,914	0,201
Расст. от эталона	0,102	0,070	0,046	0,222	0,494
ε -индикатор	0,299	0,131	0,135	0,511	0,492
Макс. ошибка	0,114	0,169	0,106	-1,028	-0,240
O_{\min}	177,163	166,061	166,783	0,063	0,060
Число решений	32,489	25,207	42,990	-0,198	0,316
Расстояние	0,537	1,529	0,505	—	—

В табл. 2 представлены средние значения показателей и относительные улучшения этих показателей для предложенных схем относительно эволюционного алгоритма. Для показателя «Расстояние между решениями» не приведены значения улучшения, так как он не отражает напрямую качество решений.

Видно, что хотя алгоритм VNMS_{long} проигрывает по некоторым показателям, двухэтапная схема VPESA превосходит эволюционные алгоритмы почти по всем показателям на 15–49%.

3.6. Дисперсия схемы. Наконец, проведены эксперименты с целью установить разброс качества получаемых решений, для чего схема была запущена 80 раз на каждом примере. В табл. 3 представлены результаты этих экспериментов. Колонка «Стандартное отклонение» содержит значение среднеквадратического отклонения для каждого параметра, усреднённого по всем примерам, а колонка «Относительное стандартное отклонение» — величину стандартного отклонения, поделённого на среднее значение показателя, также усреднённую по всем примерам.

Таблица 3

Статистики показателей качества решений

Показатель	Среднее	Станд. отклон.	Относит. станд. отклон.
Гиперобъём	0,370	0,019	0,051
Вклад	0,109	0,041	0,498
Расст. до эталона	0,012	0,003	0,266
Расст. от эталона	0,060	0,014	0,239
ε -индикатор	0,140	0,040	0,303
Макс. ошибка	0,132	0,035	0,265
O_{\min}	165,786	2,627	0,016
Число решений	42,945	2,969	0,076
Расстояние	0,515	0,142	0,429

Из табл. 3 видно, что некоторые показатели имеют небольшой разброс, в то время как другие могут значительно отклоняться от среднего значения. Наибольший разброс имеет вклад в эталонное решение, так как эта метрика зависит от точного расположения решений в пространстве целевых функций: если хоть по одной целевой функции решение чуть хуже, то оно не засчитывается как решение, вносящее вклад. Также сильно меняется расстояние между решениями, хотя число решений на Парето-границе меняется несильно. Вместе с малым изменением гиперобъёма

это может говорить о том, что основные опорные точки находятся верно, но промежуточные точки могут находиться в разных частях Парето-границы. Это же может быть причиной средних разбросов расстояний от и до эталона, ε -индикатора и максимальной ошибки.

Заключение

В работе рассмотрена новая двухкритериальная задача оптимизации «чёрного ящика» для управления потоком трафика в сети. Задача заключается в поиске такой реконфигурации весов для сетевых каналов, используемых протоколом маршрутизации, при которой число перегруженных каналов сведено к минимуму, а разница между старыми и новыми весами минимальна. Новые веса не должны увеличивать общий поток и не должны создавать новые перегруженные каналы. В ходе работы разработана специализированная схема для решения этой задачи и проведены сравнительные эксперименты для этой схемы и метаэвристических алгоритмов общего назначения.

Эксперименты показали, что алгоритм PESA2 из библиотеки MOEA Framework работает наилучшим образом среди проверенных метаэвристик, часто встречающихся в литературе. Тесты показали, что хотя упомянутые алгоритмы могут находить хорошую границу Парето, им трудно находить решения с небольшим числом перегруженных каналов, поэтому они значительно выигрывают от хороших начальных точек. Эти алгоритмы могут эффективно заполнять пробелы между заданными точками.

Обнаружено, что изменение весов по одному не позволяет найти хорошие решения с точки зрения числа перегруженных каналов, а просмотр окрестности, состоящий в изменении нескольких весов, занимает слишком большое время. Ввиду этого в работе разработана модель целочисленного линейного программирования (6)–(11) для поиска наилучшего решения в аппроксимации большой окрестности. Для этой модели вычисляются изменения нагрузки на канал после изменения одного веса. Модель ищет комбинацию из нескольких модификаций, которая минимизирует число перегруженных каналов. Затем предложен итерационный алгоритм VNMS для исходной задачи, основанный на этой модели. Этот алгоритм изменяет параметры модели для получения более качественных итоговых решений.

Алгоритм VNMS может быть использован для двух целей: найти решение с небольшим числом перегруженных каналов за короткое время или найти всё множество Парето. Во втором случае алгоритм демонстрирует хорошие результаты при решении однокритериальной задачи

минимизации числа перегруженных каналов, которые близки к результатам, полученным с помощью гибридного алгоритма на основе популяционных методов и известного алгоритма PAES. Комбинация VNMS с алгоритмами, основанными на популяции, показывает статистически лучшие на 15–49% результаты для исходной задачи, поскольку она использует преимущества обоих подходов.

Финансирование работы

Исследование выполнено в рамках государственного задания Института математики им. С. Л. Соболева (проект № FWNF–2022–0019). Дополнительных грантов на проведение или руководство этим исследованием получено не было.

Конфликт интересов

Авторы заявляют, что у них нет конфликта интересов.

Литература

1. Altın A., Fortz B., Thorup M., Ümit H. Intra-domain traffic engineering with shortest path routing protocols // Ann. Oper. Res. 2013. V. 204, No. 1. P. 65–95. DOI: 10.1007/s10479-012-1270-7.
2. Elwalid A., Jin C., Low S., Widjaja I. MATE: MPLS adaptive traffic engineering // Twenty years into the communications odyssey. Proc. 20th Annu. Joint Conf. IEEE Computer and Communications Societies (Anchorage, AK, Apr. 22–26, 2001). V. 3. Washington: IEEE Comput. Soc., 2001. P. 1300–1309. DOI: 10.1109/infcom.2001.916625.
3. Abbasi M., Guleria A., Devi M. Traffic engineering in software defined networks: A survey // J. Telecommun. Inf. Technol. 2016. V. 4. P. 3–14.
4. Bhatia R., Hao F., Kodialam M., Lakshman T. Optimized network traffic engineering using segment routing // Proc. 2015 IEEE Conf. Computer Communications (Hong Kong, China, Apr. 26–May 1, 2015). Piscataway: IEEE, 2015. P. 657–665. DOI: 10.1109/INFOCOM.2015.7218434.
5. Wang Y., Wang Z. Explicit routing algorithms for Internet traffic engineering // Proc. 8th Int. Conf. Computer Communications and Networks (Boston, MA, Oct. 11–13, 1999). Piscataway: IEEE, 1999. P. 582–588. DOI: 10.1109/ICCCN.1999.805577.
6. Poppe F., Van den Bosch S., de La Vallée-Poussin P., Van Hove H., De Neve H., Petit G. Choosing the objectives for traffic engineering in IP backbone networks based on quality-of-service requirements // Quality of future Internet service. Proc. 1st COST 263 Int. Workshop (Berlin, Germany, Sept. 25–26, 2000). Heidelberg: Springer, 2000. P. 129–140. (Lect. Notes Comput. Sci.; V. 1922). DOI: 10.1007/3-540-39939-9_11.
7. Audet C., Hare W. Derivative-free and blackbox optimization. Cham: Springer, 2017. 302 p. DOI: 10.1007/978-3-319-68913-5.

8. **Yuskov A. D., Kulachenko I. N., Melnikov A. A., Kochetov Yu. A.** Stadium antennas deployment optimization // Mathematical optimization theory and operations research. Proc. 23rd Int. Conf. MOTOR 2024 (Omsk, Russia, June 30 – July 6, 2024). Cham: Springer, 2024. P. 449–461. (Lect. Notes Comput. Sci.; V. 14766). DOI: 10.1007/978-3-031-62792-7_30.
9. **Юськов А. Д., Кулаченко И. Н., Мельников А. А., Кочетов Ю. А.** Метод декомпозиции для управления запасами в двухшелонной системе складов // Дискрет. анализ и исслед. операций. 2024. Т. 31, № 4. С. 186–212.
10. Internetworking technologies handbook. Indianapolis: Cisco Press, 2000. 1078 p.
11. **Fortz B., Thorup M.** Internet traffic engineering by optimizing OSPF weights // Reaching the Promised Land of communications. Proc. 20th Annu. Joint Conf. IEEE Computer and Communications Societies (Tel Aviv, Israel, Mar. 26–30, 2000). V. 2. Washington: IEEE Comput. Soc., 2000. P. 519–528. DOI: 10.1109/INFCOM.2000.832225.
12. **Ericsson M., Resende M., Pardalos P.** A genetic algorithm for the weight setting problem in OSPF routing // J. Comb. Optim. 2002. V. 6. P. 299–333. DOI: 10.1023/A:1014852026591.
13. **Fortz B., Thorup M.** Optimizing OSPF/IS-IS weights in a changing world // IEEE J. Sel. Areas Commun. 2002. V. 20. P. 756–767. DOI: 10.1109/JSAC.2002.1003042.
14. **Kodialam M. S., Lakshman T. V.** Network link weight setting: A machine learning based approach // Proc. IEEE Conf. Computer Communications (London, May 2–5, 2022). Piscataway: IEEE, 2022. P. 2048–2057. DOI: 10.1109/INFOCOM48880.2022.9796922.
15. **Pióro M., Szentesi Á., Harmatos J., Jüttner A., Gajowniczek P., Kozdrowski S.** On open shortest path first related network optimisation problems // Perform. Eval. 2002. V. 48, No. 1–4. P. 201–223. DOI: 10.1016/S0166-5316(02)00036-6.
16. **Balon S., Skivée F., Leduc G.** How well do traffic engineering objective functions meet TE requirements? // NETWORKING 2006. Networking technologies, services, protocols; Performance of computer and communication networks; Mobile and wireless communications systems. Proc. 5th Int. IFIP-TC6 Networking Conf. (Coimbra, Portugal, May 15–19, 2006). Heidelberg: Springer, 2006. P. 75–86. (Lect. Notes Comput. Sci.; V. 3976). DOI: 10.1007/11753810_7.
17. **Blanchy F., Melon L., Leduc G.** Routing in a MPLS network featuring pre-emption mechanisms // Proc. 10th Int. Conf. Telecommunications (Papeete, French Polynesia, Feb. 23 – Mar. 1, 2003). V. 1. Piscataway: IEEE, 2003. P. 253–260. DOI: 10.1109/ICTEL.2003.1191228.
18. **Athanasiou G., Tsagkaris K., Vlacheas P., Karvounas D., Demestichas P.** Multi-objective traffic engineering for future networks // IEEE Commun. Lett. 2012. V. 16, No. 1. P. 101–103. DOI: 10.1109/LCOMM.2011.110711.112071.

-
19. **El-Alfy E.-S. M.** Flow-based path selection for Internet traffic engineering with NSGA-II // Proc. 17th Int. Conf. Telecommunications (Doha, Qatar, Apr. 4–7, 2010). Piscataway: IEEE, 2010. P. 621–627. DOI: 10.1109/ICTEL.2010.5478839.
 20. **Sousa P., Cortez P., Rio M., Rocha M.** Traffic engineering approaches using multicriteria optimization techniques // Wired/wireless internet communications. Proc. 9th IFIP TC 6 Int. Conf. (Vilanova i la Geltrú, Spain, June 15–17, 2011). Heidelberg: Springer, 2011. P. 104–115. (Lect. Notes Comput. Sci.; V. 6649). DOI: 10.1007/978-3-642-21560-5_9.
 21. **Pereira V., Sousa P., Rocha M.** A comparison of multi-objective optimization algorithms for weight setting problems in traffic engineering // Nat. Comput. 2022. V. 21. P. 507–522. DOI: 10.1007/s11047-020-09807-1.
 22. **Wang N., Ho K. H., Pavlou G., Howarth M.** An overview of routing optimization for internet traffic engineering // IEEE Commun. Surv. Tutor. 2008. V. 10, No. 1. P. 36–56. DOI: 10.1109/COMST.2008.4483669.
 23. **Kaneda S., Uyematsu T., Nagatsu N., Sato K.** Network design and cost optimization for label switched multilayer photonic IP networks // IEEE J. Sel. Areas Commun. 2005. V. 23, No. 8. P. 1612–1619. DOI: 10.1109/JSAC.2005.851747.
 24. **Zhang-Shen R., McKeown N.** Designing a predictable Internet backbone network // Proc. 3rd Workshop Hot Topics in Networks (HotNets-III) (San Diego, CA, Nov. 15–16, 2004). New York: ACM SIGCOMM, 2004. 6 p. URL: conferences.sigcomm.org/hotnets/2004/HotNets-III%20Proceedings/zhang-shen.pdf (accessed: 10.10.2025).
 25. **Clark D. D.** Designing an Internet. Cambridge, MA: MIT Press, 2018. 432 p.
 26. **Holmberg K., Yuan D.** Optimization of internet protocol network design and routing // Networks. 2004. V. 43, No. 1. P. 39–53. DOI: 10.1002/net.10102.
 27. **Dehghani M., Vahdat V., Amiri M., Rabiei E., Salehi S.** A multi-objective optimization model for a reliable generalized flow network design // Comput. Ind. Eng. 2019. V. 138. Article ID 106074. 32 p. DOI: 10.1016/j.cie.2019.106074.
 28. **Zhu H., Gupta V., Ahuja S. S., Tian Y., Zhang Y., Jin X.** Network planning with deep reinforcement learning // Proc. ACM SIGCOMM 2021 Conf. (Delft, Netherlands, Aug. 23–28, 2021). New York: ACM, 2021. P. 258–271. DOI: 10.1145/3452296.3472902.
 29. **Wong R. T.** Telecommunications network design: Technology impacts and future directions // Networks. 2021. V. 77, No. 2. P. 205–224. DOI: 10.1002/net.21997.
 30. **Yuskov A. D., Kulachenko I. N., Melnikov A. A., Kochetov Yu. A.** Two-stage algorithm for bi-objective black-box traffic engineering // Optimization and applications. Rev. Sel. Pap. 14th Int. Conf. OPTIMA 2023 (Petrovac, Montenegro, Sept. 18–22, 2023). Cham: Springer, 2023. P. 110–125. (Lect. Notes Comput. Sci.; V. 14395). DOI: 10.1007/978-3-031-47859-8_9.

31. Hansen P., Mladenović N., Todosijević R., Hanafi S. Variable neighborhood search: Basics and variants // *EURO J. Comput. Optim.* 2016. V. 5, No. 3. P. 423–454. DOI: 10.1007/s13675-016-0075-x.
32. Forrest J., Ralphs T., Vigerske S. [et al.]. COIN-OR Branch-and-Cut solver. Genève: CERN; Zenodo, 2023. DOI: 10.5281/zenodo.10041724.
33. Audet C., Bégin J., Cartier D., Le Digabel S., Salomon L. Performance indicators in multiobjective optimization // *Eur. J. Oper. Res.* 2021. V. 292, No. 2. P. 397–422. DOI: 10.1016/j.ejor.2020.11.016.
34. Konak A., Coit D. W., Smith A. E. Multi-objective optimization using genetic algorithms: A tutorial // *Reliab. Eng. Syst. Saf.* 2006. V. 91, No. 9. P. 992–1007. DOI: 10.1016/j.ress.2005.11.018.
35. Hadka D. MOEA framework: A free and open source Java framework for multiobjective optimization. Version 5.1. 2025. URL: moeaframework.org (accessed: 10.10.2025).
36. Deb K., Pratap A., Agarwal S., Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II // *IEEE Trans. Evolut. Comput.* 2002. V. 6, No. 2. P. 182–197. DOI: 10.1109/4235.996017.
37. Zitzler E., Laumanns M., Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. Inst. Tech. Inform. Kommun. rep. 103. Zürich: ETH Zürich, 2001. DOI: 10.3929/ethz-a-004284029.
38. Corne D. W., Jerram N. R., Knowles J. D., Oates M. J. PESA-II: Region-based selection in evolutionary multiobjective optimization // Proc. 3rd Annu. Conf. Genetic and Evolutionary Computation (San Francisco, CA, July 7–11, 2001). San Francisco: Morgan Kaufmann Publ., 2001. P. 283–290.
39. Knowles J. D., Corne D. W. Approximating the nondominated front using the Pareto archived evolution strategy // *Evolut. Comput.* 2000. V. 8, No. 2. P. 149–172. DOI: 10.1162/106365600568167.
40. Mann H. B., Whitney D. R. On a test of whether one of two random variables is stochastically larger than the other // *Ann. Math. Stat.* 1947. V. 18, No. 1. P. 50–60. DOI: 10.1214/aoms/1177730491.
41. Wilcoxon F. Individual comparisons by ranking methods // *Biom. Bull.* 1945. V. 1, No. 6. P. 80–83.
42. Lindauer M., Eggensperger K., Feurer M. [et al.]. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization // *J. Mach. Learn. Res.* 2022. V. 23. Article ID 54. 9 p.

Юськов Александр Дмитриевич
Кулаченко Игорь Николаевич
Мельников Андрей Андреевич
Кочетов Юрий Андреевич

Статья поступила
3 февраля 2025 г.
После доработки —
23 мая 2025 г.
Принята к публикации
22 июня 2025 г.

A HYBRID ALGORITHM FOR A TWO-OBJECTIVE TRAFFIC ENGINEERING PROBLEM

A. D. Yuskov^{1, a}, I. N. Kulachenko^{2, b},
 A. A. Melnikov^{2, c}, and Y. A. Kochetov^{2, d}

¹ Novosibirsk State University,
 2 Pirogov Street, 630090 Novosibirsk, Russia

² Sobolev Institute of Mathematics,
 4 Acad. Koptyug Avenue, 630090 Novosibirsk, Russia
 E-mail: ^aa.yuskov@g.nsu.ru, ^bink@math.nsc.ru,
^cmelnikov@math.nsc.ru, ^djkochet@math.nsc.ru

Abstract. We consider an Internet traffic routing problem. The paths for requests are assigned implicitly by setting link weights. The loads of links are generated by a simulator. If the load of a link is greater than its capacity, then the link is called congested. Our goal is to minimize two objective functions: the number of congested links and the distance between the initial and current weight vectors. The problem also includes two constraints: the total link flow in the network has an upper bound and new congested links are unwanted. We propose a new two-stage evolutionary scheme. The scheme employs a local search algorithm with a large neighbourhood to find an initial approximation of the Pareto set. The algorithm utilizes an integer linear programming model to determine the best solution in the neighbourhood. We compare the proposed scheme with well-known evolutionary algorithms using instances with 628 links and 1324 requests. According to the experiments, the proposed scheme constructs solutions statistically better at 15–49% for many performance indicators (9 out of 10). Tab. 3, illustr. 6, bibliogr. 42.

Keywords: black box optimization, matheuristic, variable neighbourhood search, OSPF, evolutionary algorithm.

References

1. **A. Altın, B. Fortz, M. Thorup, and H. Ümit**, Intra-domain traffic engineering with shortest path routing protocols, *Ann. Oper. Res.* **204** (1), 65–95 (2013), DOI: 10.1007/s10479-012-1270-7.

2. **A. Elwalid, C. Jin, S. Low, and I. Widjaja**, MATE: MPLS adaptive traffic engineering, in *Twenty Years into the Communications Odyssey*, Proc. 20th Annu. Joint Conf. IEEE Computer and Communications Societies (Anchorage, AK, Apr. 22–26, 2001), Vol. 3 (IEEE Comput. Soc., Washington, 2001), pp. 1300–1309, DOI: 10.1109/infcom.2001.916625.
3. **M. Abbasi, A. Guleria, and M. Devi**, Traffic engineering in software defined networks: A survey, *J. Telecommun. Inf. Technol.* **4**, 3–14 (2016).
4. **R. Bhatia, F. Hao, M. Kodialam, and T. Lakshman**, Optimized network traffic engineering using segment routing, in *Proc. 2015 IEEE Conf. Computer Communications* (Hong Kong, China, Apr. 26–May 1, 2015) (IEEE, Piscataway, 2015), pp. 657–665, DOI: 10.1109/INFOCOM.2015.7218434.
5. **Y. Wang and Z. Wang**, Explicit routing algorithms for Internet traffic engineering, in *Proc. 8th Int. Conf. Computer Communications and Networks* (Boston, MA, Oct. 11–13, 1999) (IEEE, Piscataway, 1999), pp. 582–588, DOI: 10.1109/ICCCN.1999.805577.
6. **F. Poppe, S. Van den Bosch, P. de La Vallée-Poussin, H. Van Hove, H. De Neve, and G. Petit**, Choosing the objectives for traffic engineering in IP backbone networks based on quality-of-service requirements, in *Quality of Future Internet Service*, Proc. 1st COST 263 Int. Workshop (Berlin, Germany, Sept. 25–26, 2000) (Springer, Heidelberg, 2000), pp. 129–140 (Lect. Notes Comput. Sci., Vol. 1922), DOI: 10.1007/3-540-39939-9_11.
7. **C. Audet and W. Hare**, *Derivative-Free and Blackbox Optimization* (Springer, Cham, 2017), DOI: 10.1007/978-3-319-68913-5.
8. **A. D. Yuskov, I. N. Kulachenko, A. A. Melnikov, and Yu. A. Kochetov**, Stadium antennas deployment optimization, in *Mathematical Optimization Theory and Operations Research*, Proc. 23rd Int. Conf. MOTOR 2024 (Omsk, Russia, June 30–July 6, 2024) (Springer, Cham, 2024), pp. 449–461 (Lect. Notes Comput. Sci., Vol. 14766), DOI: 10.1007/978-3-031-62792-7_30.
9. **A. D. Yuskov, I. N. Kulachenko, A. A. Melnikov, and Yu. A. Kochetov**, Decomposition approach for a two-echelon inventory management system, *Diskretn. Anal. Issled. Oper.* **31** (4), 186–212 (2024), DOI: 10.33048/daio.2024.31.794 [Russian] [*J. Appl. Ind. Math.* **18** (4) 918–934 (2024)], DOI: 10.1134/S1990478924040239.
10. *Internetworking Technologies Handbook* (Cisco Press, Indianapolis, 2000).
11. **B. Fortz and M. Thorup**, Internet traffic engineering by optimizing OSPF weights, in *Reaching the Promised Land of Communications*, Proc. 20th Annu. Joint Conf. IEEE Computer and Communications Societies (Tel Aviv, Israel, Mar. 26–30, 2000), Vol. 2 (IEEE Comput. Soc., Washington, 2000), pp. 519–528, DOI: 10.1109/INFCOM.2000.832225.
12. **M. Ericsson, M. Resende, and P. Pardalos**, A genetic algorithm for the weight setting problem in OSPF routing, *J. Comb. Optim.* **6**, 299–333 (2002), DOI: 10.1023/A:1014852026591.
13. **B. Fortz and M. Thorup**, Optimizing OSPF/IS-IS weights in a changing world, *IEEE J. Sel. Areas Commun.* **20**, 756–767 (2002), DOI: 10.1109/JSAC.2002.1003042.

14. **M. S. Kodialam and T. V. Lakshman**, Network link weight setting: A machine learning based approach, in *Proc. IEEE Conf. Computer Communications* (London, May 2–5, 2022) (IEEE, Piscataway, 2022), pp. 2048–2057, DOI: 10.1109/INFOCOM48880.2022.9796922.
15. **M. Pióro, Á. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, and S. Kozdrowski**, On open shortest path first related network optimisation problems, *Perform. Eval.* **48** (1–4), 201–223 (2002), DOI: 10.1016/S0166-5316(02)00036-6.
16. **S. Balon, F. Skivée, and G. Leduc**, How well do traffic engineering objective functions meet TE requirements?, in *NETWORKING 2006. Networking Technologies, Services, Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, Proc. 5th Int. IFIP-TC6 Networking Conf. (Coimbra, Portugal, May 15–19, 2006) (Springer, Heidelberg, 2006), pp. 75–86 (Lect. Notes Comput. Sci., Vol. 3976), DOI: 10.1007/11753810_7.
17. **F. Blanchy, L. Melon, and G. Leduc**, Routing in a MPLS network featuring preemption mechanisms, in *Proc. 10th Int. Conf. Telecommunications* (Papeete, French Polynesia, Feb. 23–Mar. 1, 2003), Vol. 1 (IEEE, Piscataway, 2003), pp. 253–260, DOI: 10.1109/ICTEL.2003.1191228.
18. **G. Athanasiou, K. Tsagkaris, P. Vlacheas, D. Karvounas, and P. Demestichas**, Multi-objective traffic engineering for future networks, *IEEE Commun. Lett.* **16** (1), 101–103 (2012), DOI: 10.1109/LCOMM.2011.110711.112071.
19. **E.-S. M. El-Alfy**, Flow-based path selection for Internet traffic engineering with NSGA-II, in *Proc. 17th Int. Conf. Telecommunications* (Doha, Qatar, Apr. 4–7, 2010) (IEEE, Piscataway, 2010), pp. 621–627, DOI: 10.1109/ICTEL.2010.5478839.
20. **P. Sousa, P. Cortez, M. Rio, and M. Rocha**, Traffic engineering approaches using multicriteria optimization techniques, in *Wired/Wireless Internet Communications*, Proc. 9th IFIP TC 6 Int. Conf. (Vilanova i la Geltrú, Spain, June 15–17, 2011) (Springer, Heidelberg, 2011), pp. 104–115 (Lect. Notes Comput. Sci., Vol. 6649), DOI: 10.1007/978-3-642-21560-5_9.
21. **V. Pereira, P. Sousa, and M. Rocha**, A comparison of multi-objective optimization algorithms for weight setting problems in traffic engineering, *Nat. Comput.* **21**, 507–522 (2022), DOI: 10.1007/s11047-020-09807-1.
22. **N. Wang, K. H. Ho, G. Pavlou, and M. Howarth**, An overview of routing optimization for internet traffic engineering, *IEEE Commun. Surv. Tutor.* **10** (1), 36–56 (2008), DOI: 10.1109/CST.2008.4483669.
23. **S. Kaneda, T. Uyematsu, N. Nagatsu, and K. Sato**, Network design and cost optimization for label switched multilayer photonic IP networks, *IEEE J. Sel. Areas Commun.* **23** (8), 1612–1619 (2005), DOI: 10.1109/JSAC.2005.851747.
24. **R. Zhang-Shen and N. McKeown**, Designing a predictable Internet backbone network, in *Proc. 3rd Workshop Hot Topics in Networks (HotNets-III)* (San Diego, CA, Nov. 15–16, 2004) (ACM SIGCOMM, New

- York, 2004), URL: conferences.sigcomm.org/hotnets/2004/HotNets-III%20Proceedings/zhang-shen.pdf (accessed: 10.10.2025).
- 25. **D. D. Clark**, *Designing an Internet* (MIT Press, Cambridge, MA, 2018).
 - 26. **K. Holmberg** and **D. Yuan**, Optimization of internet protocol network design and routing, *Networks* **43** (1), 39–53 (2004), DOI: [10.1002/net.10102](https://doi.org/10.1002/net.10102).
 - 27. **M. Dehghani**, **V. Vahdat**, **M. Amiri**, **E. Rabiei**, and **S. Salehi**, A multi-objective optimization model for a reliable generalized flow network design, *Comput. Ind. Eng.* **138**, ID 106074 (2019), DOI: [10.1016/j.cie.2019.106074](https://doi.org/10.1016/j.cie.2019.106074).
 - 28. **H. Zhu**, **V. Gupta**, **S. S. Ahuja**, **Y. Tian**, **Y. Zhang**, and **X. Jin**, Network planning with deep reinforcement learning, in *Proc. ACM SIGCOMM 2021 Conf.* (Delft, Netherlands, Aug. 23–28, 2021) (ACM, New York, 2021), pp. 258–271, DOI: [10.1145/3452296.3472902](https://doi.org/10.1145/3452296.3472902).
 - 29. **R. T. Wong**, Telecommunications network design: Technology impacts and future directions, *Networks* **77** (2), 205–224 (2021), DOI: [10.1002/net.21997](https://doi.org/10.1002/net.21997).
 - 30. **A. D. Yuskov**, **I. N. Kulachenko**, **A. A. Melnikov**, and **Yu. A. Kochetov**, Two-stage algorithm for bi-objective black-box traffic engineering, in *Optimization and Applications*, Rev. Sel. Pap. 14th Int. Conf. OPTIMA 2023 (Petrovac, Montenegro, Sept. 18–22, 2023) (Springer, Cham, 2023), pp. 110–125 (Lect. Notes Comput. Sci., Vol. 14395), DOI: [10.1007/978-3-031-47859-8_9](https://doi.org/10.1007/978-3-031-47859-8_9).
 - 31. **P. Hansen**, **N. Mladenović**, **R. Todosijević**, and **S. Hanafi**, Variable neighborhood search: Basics and variants, *EURO J. Comput. Optim.* **5** (3), 423–454 (2016), DOI: [10.1007/s13675-016-0075-x](https://doi.org/10.1007/s13675-016-0075-x).
 - 32. **J. Forrest**, **T. Ralphs**, **S. Vigerske**, [et al.], COIN-OR Branch-and-Cut solver (CERN; Zenodo, Genève, 2023), DOI: [10.5281/zenodo.10041724](https://doi.org/10.5281/zenodo.10041724).
 - 33. **C. Audet**, **J. Bigeon**, **D. Cartier**, **S. Le Digabel**, and **L. Salomon**, Performance indicators in multiobjective optimization, *Eur. J. Oper. Res.* **292** (2), 397–422 (2021), DOI: [10.1016/j.ejor.2020.11.016](https://doi.org/10.1016/j.ejor.2020.11.016).
 - 34. **A. Konak**, **D. W. Coit**, and **A. E. Smith**, Multi-objective optimization using genetic algorithms: A tutorial, *Reliab. Eng. Syst. Saf.* **91** (9), 992–1007 (2006), DOI: [10.1016/j.ress.2005.11.018](https://doi.org/10.1016/j.ress.2005.11.018).
 - 35. **D. Hadka**, MOEA framework: A free and open source Java framework for multiobjective optimization. Version 5.1 (2025), URL: moeaframework.org (accessed: 10.10.2025).
 - 36. **K. Deb**, **A. Pratap**, **S. Agarwal**, and **T. Meyarivan**, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evolut. Comput.* **6** (2), 182–197 (2002), DOI: [10.1109/4235.996017](https://doi.org/10.1109/4235.996017).
 - 37. **E. Zitzler**, **M. Laumanns**, and **L. Thiele**, SPEA2: Improving the strength Pareto evolutionary algorithm, *Inst. Tech. Inform. Kommun. rep.* **103** (ETH Zürich, Zürich, 2001), DOI: [10.3929/ethz-a-004284029](https://doi.org/10.3929/ethz-a-004284029).
 - 38. **D. W. Corne**, **N. R. Jerram**, **J. D. Knowles**, and **M. J. Oates**, PESA-II: Region-based selection in evolutionary multiobjective optimization, in *Proc. 3rd Annu. Conf. Genetic and Evolutionary Computation* (San Francisco, CA, July 7–11, 2001) (Morgan Kaufmann Publ., San Francisco, 2001), pp. 283–290.

-
- 39. **J. D. Knowles** and **D. W. Corne**, Approximating the nondominated front using the Pareto archived evolution strategy, *Evolut. Comput.* **8** (2), 149–172 (2000), DOI: 10.1162/106365600568167.
 - 40. **H. B. Mann** and **D. R. Whitney**, On a test of whether one of two random variables is stochastically larger than the other, *Ann. Math. Stat.* **18** (1), 50–60 (1947), DOI: 10.1214/aoms/1177730491.
 - 41. **F. Wilcoxon**, Individual comparisons by ranking methods, *Biom. Bull.* **1** (6), 80–83 (1945).
 - 42. **M. Lindauer**, **K. Eggensperger**, **M. Feurer**, [et al.], SMAC3: A versatile Bayesian optimization package for hyperparameter optimization, *J. Mach. Learn. Res.* **23**, ID 54 (2022).

Aleksandr D. Yuskov

Received February 3, 2025

Igor N. Kulachenko

Revised May 23, 2025

Andrey A. Melnikov

Accepted June 22, 2025

Yury A. Kochetov