

ДИФФЕРЕНЦИАЛЬНЫЕ
УРАВНЕНИЯ
И

ПРОЦЕССЫ УПРАВЛЕНИЯ

№ 3, 2025

Электронный журнал,
рег. Эл № ФС77-39410 от 15.04.2010
ISSN 1817-2172

<http://diffjournal.spbu.ru/>
e-mail: jodiff@mail.ru

Общая теория управления

Разработка гибридного алгоритма многокритериальной оптимизации для восстановления свойств пористой среды по фильтрационным экспериментам

Цыкунов О.И.^{1,2,*}

¹ ООО «Газпромнефть-НТЦ», Санкт-Петербург, Российская Федерация

² Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Российская Федерация

* olegtsykunovmgdn@gmail.com

Аннотация. Рассматривается задача определения физических свойств пористой среды по данным фильтрационных экспериментов. Используется двухкомпонентная модель капиллярной пропитки в среде с двойной пористостью в виде системы трех дифференциальных уравнений, которая содержит 10 свободных параметров. Для автоматизации их определения на основе серии из трех экспериментов используются алгоритмы оптимизации. Соответствие модели каждому эксперименту является целевой функцией, поэтому рассматривается задача многокритериальной оптимизации. Для тестирования выбраны мета-эвристические алгоритмы – NSGA-II, NSGA-III, SMPSO, OMOPSO, MOEA/D. Для каждого алгоритма проведена Байесовская мета-оптимизация гиперпараметров для 5 тестовых функций: DTLZ1, DTLZ2, MW8, WFG1, WFG2. Определено, что наиболее эффективными алгоритмами по точности и затраченному вычислительному времени являются SMPSO и NSGA-III. В задачах с большей вычислительной сложностью более эффективен SMPSO, а в простых задачах – NSGA-III. Для получения эффективного во всех тестовых функция метода оптимизации создан гибридный алгоритм. Протестированы различные методы гибридизации, лучшим вариантом стала гибридизация по типу процессор/постпроцессор, где сначала иницируется SMPSO, после чего популяция передается в NSGA-III. Для улучшения гибридного алгоритма применен метод кластеризации K-средних для отбора из популяции SMPSO наиболее перспективных кластеров частиц с целью передачи их в NSGA-III. Также введен адаптивный критерий остановки алгоритмов на основе отслеживания динамики изменения метрик получаемых решений – H-value и Spacing. Сопоставление полученного алгоритма с роевыми

и генетическими алгоритмами показало, что предложенный алгоритм за одно и то же количество итераций находит сочетание свободных параметров с меньшей невязкой между моделью и экспериментом.

Ключевые слова: многокритериальная оптимизация, метод роя частиц, генетический алгоритм, мета-оптимизация алгоритма, модель капиллярной пропитки, NSGA-III, SMPSO

Введение

Процесс моделирования фильтрации в нефтенасыщенных пористых средах связан с высокой неопределенностью физических свойств горных пород. Это связано с невозможностью прямого исследования параметров породы, залегающей на глубине в несколько километров. При проведении экспериментов на керне определить все его свойства сложно из-за больших финансовых и временных затрат. В большинстве случаев, исследования ограничиваются стандартным набором измерений пористости, абсолютной и фазовой проницаемостей [1]. Специальные исследования проводятся редко, а весь комплекс не проводится на одном и том же образце. Следовательно, при моделировании фильтрации используются осредненные параметры пласта [2]. При решении фильтрационных задач, например, в случае капиллярной пропитки, существует недостаток исходной информации, так как процесс зависит от микроструктуры пористой среды. Вследствие этого модель пропитки имеет множество свободных параметров, определение которых необходимо как для настройки модели, так и для ее использования в прогнозировании технологических процессов [2].

Для автоматизации процесса определения набора неизвестных свойств среды в данной работе предлагается использовать оптимизационные алгоритмы, которые могут определить такое сочетание параметров, при котором невязки между серией экспериментов и результатами моделирования будут минимальны. Поэтому целью данной работы является разработка эффективного алгоритма оптимизации с целью определения свободных параметров модели капиллярной пропитки на основе фильтрационных экспериментов на керне.

Для подобных задач часто применяются специализированные алгоритмы многокритериальной оптимизации [3], исследованием которых занималось большое количество авторов. Алгоритм многокритериального роя частиц MOPSO был разработан М. Р. Сьеррой и К.А. Козльо [4] на основе одноцелевого роя частиц PSO, разработанного Д. Кеннеди и Р. Еберхартом [5]. Роевые алгоритмы хорошо проявляют себя в поиске глобальных оптимумов [3], когда как для поиска локального оптимума в более узкой области решений применяются генетические алгоритмы. Их исследовали К. Деб [6], Г. Джейн [7], Д. Бланк [8], большой вклад также внесли работы А.К. Гарагуловой [9], И. Даса и Д.Е. Денниса [10].

Чтобы получить наиболее эффективный алгоритм с наибольшей точностью и наименьшим временем работы используется гибридизация алгоритмов, когда несколько различных методов запускаются последовательно или параллельно [3], что позволяет совместить преимущества разных алгоритмов. Ранее исследователи уже комбинировали роевые алгоритмы глобального поиска и локальные генетические, например, в работе [11] предлагается последовательный гибрид GA и PSO. В работах [12,13] предлагаются параллельные гибридные алгоритмы. Основной проблемой при таком подходе является передача популяции из одного алгоритма в другой, так как перенос неудачных решений замедляет скорость оптимизации. Данную проблему ранее решали с помощью различных методов сортировки решений, например, в работе [14] использовался алгоритм “жадного отбора”. В данном исследовании предлагается использовать гибридный последовательный многокритериальный алгоритм оптимизации с использованием алгоритма кластеризации при передаче популяции с целью выделения наиболее близких кластеров решений к идеальной

точке (точка, в которой все целевые функции равны нулю), что позволяет значительно увеличить эффективность алгоритма. Ранее алгоритмы кластеризации не применялись для фильтрации популяции в гибридных алгоритмах. Также предлагается использование адаптивных критериев остановки алгоритма на основе отслеживания динамики изменения различных характеристик полученных решений.

1. Математическая модель фильтрации

Развитие современных технологий позволяет вводить в разработку месторождения с низкой проницаемостью, на которых капиллярные эффекты оказывают существенное влияние на фильтрацию. К ним относится процесс капиллярной пропитки – самопроизвольное проникновение смачивающей фазы в пористую среду и вытеснение несмачивающей [2].

Ранее в работе [2] была предложена двухкомпонентная модель капиллярной пропитки, основанная на одномерной модели Кая [15] и теории двойной среды. Модель описывает фильтрацию жидкости под действием внешнего градиента давления и капиллярных сил. В ней также учитывается неоднородность порового пространства и неравномерность распределения капиллярных сил с помощью разделения породы на две среды с различными свойствами. Это позволило точно воспроизвести серию экспериментов из работы Танга и Фирузабади [16]. При этом модель не требует проведения экспериментов по измерению капиллярного давления, так как оно определяется по уравнению Лапласа (1) из свойств горной породы и размера поровых каналов:

$$P_c = \frac{2\sigma \cos \theta}{\alpha r}, \quad (1)$$

где P_c – капиллярное давление, σ – коэффициент поверхностного натяжения, r – радиус порового канала, θ – контактный угол смачивания, α – коэффициент кривизны сечения поры [2]. Фильтрация в поровом канале описывается уравнением Хагена-Пуазейля:

$$q = \frac{\pi(\alpha r)^4 \Delta P}{8\mu L_a}, \quad (2)$$

где q – дебит жидкости, ΔP – перепад давления, $\Delta P = P_c + P_{inj}$, P_{inj} – давление закачки, μ – вязкость жидкости, L_a – длина порового канала.

Модель представляет собой систему дифференциальных уравнений первого порядка, описывает фильтрацию в двойной среде под действием градиента давления и капиллярных сил и учитывает массообмен между ними:

$$\begin{cases} \frac{dM_1(t)}{dt} = k_1^2 \frac{z_1}{M_1(t)} + \frac{dM_3(t)}{dt} \\ \frac{dM_2(t)}{dt} = k_2^2 \frac{z_2}{M_2(t)} - \frac{dM_3(t)}{dt} \\ \frac{dM_3(t)}{dt} = k_3^2 \frac{z_3}{M_3(t)} n \left(\frac{M_2(t)}{k_2} - \frac{M_1(t)}{k_1} \right) \end{cases}, \quad (3)$$

где 1 – среда с малыми порами, 2 – среда с большими порами, 3 – промежуточные капилляры между средами, t – время, $M(t)$ – накопленная масса воды, n – плотность промежуточных капилляров между средами 1 и 2 на единицу длины. Коэффициенты k_i и z_i определяются как:

$$k_i = \rho A_i \varphi_i (S_{wf} - S_{wi}), \quad (4)$$

где i – номер среды, ρ – плотность воды, A – площадь фильтрации, S_{wi} – начальная (связанная) водонасыщенность, S_{wf} – конечная водонасыщенность, φ – пористость.

$$z_i = \frac{\alpha^3 r_i \sigma \cos \theta}{4 \mu \tau^2} + \frac{\alpha^4 r_i^2 P_{inj}}{8 \mu \tau^2}, \quad (5)$$

где τ – коэффициент кривизны порового канала.

Начальные условия:

$$M_1(0) = 0; M_2(0) = 0; M_3(0) = 0. \quad (6)$$

В начальных условиях (6) содержится нулевое значение массы в начальный момент времени из-за чего в решении возникает деление на ноль [2]. Для решения данной проблемы на начальном участке пропитки при $t < 1$ с принято допущение, что массообмен равен нулю, тогда в начальном промежутке времени система вырождается в два независимых уравнения и каждая среда считается отдельно друг от друга [2].

Модель имеет 10 свободных параметров, которые сложно определить в лабораторных условиях: параметры кривизны τ и формы a поровых каналов, параметры промежуточной среды, соотношение между радиусами и пористостями среды u при известных общих среднем радиусе и средней пористости. Помимо этого, параметры смачиваемости породы σ и θ зачастую не определяются при проведении экспериментов, либо определены неточно. Из-за этого появляется необходимость в подборе такой комбинации параметров, при которой модель будет воспроизводить фактические кривые фильтрации. Ручной подбор 10 переменных требует больших временных затрат, поэтому для автоматизации данной задачи предлагается использовать оптимизационные алгоритмы.

2. Задача оптимизации

Существуют различные подходы для определения свободных параметров. Самым простым методом является использование месторождений-аналогов, на которых неизвестные свойства были определены. Но некоторые параметры измерить сложно, например – кривизну и форму пор. В таком случае необходимо подбирать комбинацию свойств, при которой модель согласуется с экспериментом. Если свободных параметров достаточно много, то необходимо использовать методы оптимизации, где исходными параметрами будут являться параметры породы, а целевыми функциями – отклонение модели от экспериментов. Так как в исследовании [16] серия состоит из трех экспериментов, то минимизировать необходимо три целевых функции отклонений. Тогда необходимо решить задачу многокритериальной оптимизации с тремя целевыми функциями и 10 переменными:

$$\min\{f_1(\bar{X}), f_2(\bar{X}), f_3(\bar{X})\} = f_1(\bar{X}^*), f_2(\bar{X}^*), f_3(\bar{X}^*), \quad (7)$$

где $f_k(\bar{X})$ – целевая функция для каждого из трех экспериментов, \bar{X} – вектор варьируемых параметров, \bar{X}^* – искомые оптимальные решения, 1, 2, 3 – номер эксперимента. Зададим целевую функцию:

$$f_k(\bar{X}) = \sum_{i=1}^n |M_i^{k,exp} - M_i^{k,mod}(\bar{X})|, \quad (8)$$

где n – количество временных шагов, i – текущий шаг по времени, M^{exp} – масса пропитки в эксперименте, M^{mod} – в модели.

Для решения задачи необходимо определить диапазоны изменения настраиваемых параметров. В Таблице 1 приведен список переменных и их диапазоны. Модель имеет 10 свободных параметров, 4 – фиктивные и описывают интенсивность массообмена, 4 – могут определяться стандартными лабораторными исследованиями, 2 – могут быть определены только на уникальных установках по микро-сканированию зерна. Значения параметров 1-5 определялись исходя из физических пределов и лабораторных данных, для параметров 6-10 проводился анализ чувствительности модели для выявления диапазонов значений.

Таб. 1. Варьируемые параметры и диапазоны их изменения

№	Параметр	Ед. изм	Описание	Мин. значение	Макс. значение
1	σ	Поверхностное натяжение	Н · м	0.025	0.03
2	θ	Угол смачивания	°град	0	90
3	φ_2	Пористость среды 2	д.ед.	0.05	0.25
4	τ	Параметр кривизны каналов	д.ед.	1	20
5	a	Параметр формы каналов	д.ед.	1	2.5
6	φ_3	Пористость среды 3	д.ед.	0.05	0.25
7	n	Количество промежуточных пор среды 3 на единицу длины	1/м	$5 \cdot 10^3$	$5 \cdot 10^5$
8	A_3	Площадь контакта среды 3	м ²	$1 \cdot 10^{-8}$	$1 \cdot 10^{-5}$
9	y	Коэффициент, для определения r_1	д.ед.	1	5
10	r_3	Радиус промежуточных пор	м	$1 \cdot 10^{-8}$	$1 \cdot 10^{-6}$

3. Выбор тестовых функций

Задача оптимизации является многокритериальной, поэтому необходимо рассматривать не единичное оптимальное решение, а набор решений эффективных по Парето [3]. Такими являются решения, в которых ни одна целевая функция не может быть улучшена без ухудшения другой целевой функции [3, 17]. Для трех целевых функций должен существовать трехмерный фронт Парето-оптимальных решений, из которого можно выбрать одно решение, в зависимости от того, какие эксперименты наиболее важны.

Решение системы дифференциальных уравнений (3) реализовано на языке программирования Python с использованием библиотеки SciPy и метода Рунге-Кутты четвертого порядка. Для выбора эффективного алгоритма требуется большое число расчетов, поэтому предлагается заменить исходную задачу на набор более простых тестовых функций. Такой подход используется для анализа эффективности оптимизационных алгоритмов [3].

В первую очередь был определен истинный Парето-фронт задачи, чтобы выбрать тестовые задачи с аналогичными формами фронтов. Для этого использовались диапазоны переменных из Таблицы 1 и метод латинского гиперкуба (LHS) [18]. Преимуществом данного метода является более равномерное покрытие выборки точек в сравнении со

случайной выборкой. В итоге был получен плоский фронт Парето-оптимальных решений, приведенный на Рис. 1.

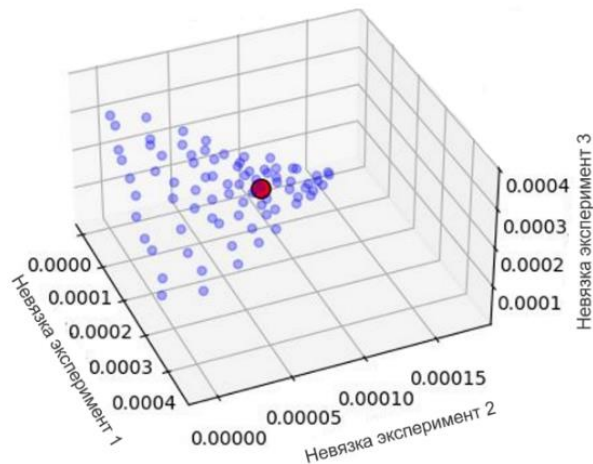
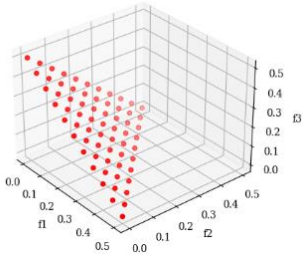
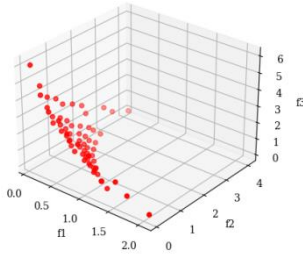
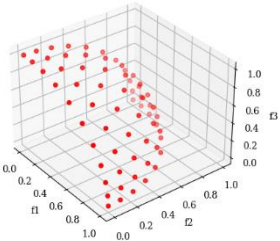
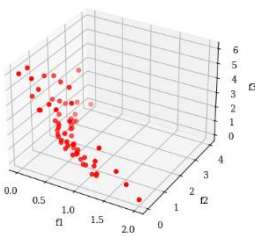
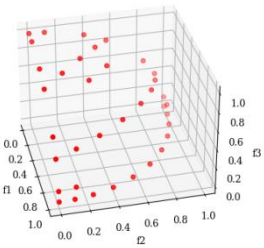
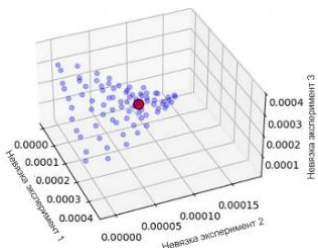


Рис. 1. Предполагаемая форма Парето-фронта для рассматриваемой задачи: синим – точки Парето-фронта, красным – ближайшая к идеальной точке

Таб. 2. Тестовые функции многокритериальной оптимизации

Тестовая функция	Вид Парето-фронта	Тестовая функция	Вид Парето-фронта
DTLZ1		WFG1	
DTLZ2		WFG2	
MW8		Исходный	

Существует большое количество тестовых функций для оптимизационных задач с тремя целевыми функциями. Для тестирования оптимизационных алгоритмов часто используются задачи из серии DTLZ (Deb, Thiele, Laumanns, Zitzler) [19], MW (Ma, Wang) [20] и WFG (Walking Fish Group) [21]. Из них было выбрано 5 функций – DTLZ1, DTLZ2, MW8, WFG1,

WFG2. Наиболее близкую форму Парето-фронта к рассматриваемой задаче имеют DTLZ1 и WFG1-2. Истинные Парето-фронты тестовых функций приведены в Таблице 2.

4. Алгоритмы многоцелевой оптимизации

В решении задач многокритериальной оптимизации для поиска глобального оптимума наиболее широко применяются мета-эвристические алгоритмы – генетические, популяционные и эволюционные, так как их эффективность по затраченному времени и качеству решения превосходит остальные виды алгоритмов в условиях большого количества целевых функций и переменных [3]. В данной работе будут рассмотрены 5 наиболее распространенных методов оптимизации: генетические алгоритмы NSGA-II, NSGA-III, методы роя частиц SMPSO, OMOPSO, эволюционный алгоритм MOEA/D. Именно эти алгоритмы применяются наиболее часто, что подтверждается реализацией их в специализированных оптимизационных библиотеках, например, в Rmoo [8].

NSGA-II

Алгоритм многокритериальной оптимизации NSGA-II (Non-dominated Sorting Genetic Algorithm II) является эволюционным алгоритмом, который использует механизмы естественного отбора, мутации и рекомбинации для поиска оптимальных решений [22]. Задача заключается в минимизации нескольких целевых функций $f_1(\bar{X}), f_2(\bar{X}), \dots, f_m(\bar{X})$. Решение \bar{X}^* считается доминирующим по отношению к решению \bar{Y}^* , если \bar{X}^* лучше или не хуже, чем \bar{Y}^* по всем критериям и лучше хотя бы по одному. Алгоритм базируется на сортировке популяции на группы не доминирующих решений (Fast Non-Dominated Sort - FNDS) и на оценке скученности решений (Crowding Distance-Assignment - CDA) [9]. С помощью FNDS определяется ранг особи r . Решения с одинаковым рангом r объединяются в фронты [22]. NSGA-II включает в себя этапы: инициализация популяции - создается произвольная популяция P размерности N . Далее происходит оценка целевых функций, процедура сортировки (FNDS), оценка скученности решение, расчет CDA. Выбираются лучшие решения, которые имеют меньший ранг r , применяются SBX-кроссовер [23] и мутация. Итерации продолжаются, пока не будет достигнуто максимальное количество поколений.

NSGA-III

Алгоритм NSGA-III является развитием NSGA-II и предназначен для решения задач многокритериальной оптимизации, когда количество целевых функций больше двух. Он направлен на улучшение разнообразия решений на сложных Парето-фронтах [6, 7]. Главным отличием от NSGA-II является сортировка на основе опорных направлений: каждый индивид получает соответствующее направление в многомерном пространстве. Опорные направления используются для задания начального положения частиц. Для генерации опорных направлений наиболее часто используется структурированный подход Даса и Денниса [10].

OMOPSO

Алгоритм OMOPSO является развитием многокритериального алгоритма роя частиц MOPSO (Multi-Objective Particle Swarm Optimization) [4]. Базовый алгоритм роя частиц (PSO) был введен Д. Кеннеди и Р. Эберхартом [5]. Основной принцип алгоритма – инициация набора (роя) частиц, каждая частица представляет собой потенциальное решение и имеет положение x_i и скорость v_i . Скорость и позиция частицы обновляется на основе следующих формул:

$$v_i(t+1) = wv_i(t) + c_1r_1(p_i^{best} - x_i) + c_2r_2(G^{best} - x_i), \quad (9)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (10)$$

где w — инерционный вес, c_1 и c_2 — коэффициенты когнитивного и социального обучения, r_1 и r_2 — случайные числа в диапазоне $[0;1]$, P_i^{best} — лучшая позиция частицы, а G^{best} — глобальная лучшая позиция. Если текущее значение целевой функции для данной частицы лучше, чем ее P_i^{best} — обновляется P_i^{best} . Если текущее значение лучше, чем G^{best} — обновляется G^{best} . Процесс повторяется, пока не будет достигнуто предельное количество итераций или достаточная сходимость. MOPSO расширяет классический PSO и учитывает несколько целевых функций одновременно. Помимо этого, аналогично алгоритму NSGA-II каждая частица помещается в не доминирующее множество, что позволяет отслеживать наилучшие решения по множеству критериев. Каждая частица обновляет свою скорость и позицию аналогично классическому PSO, но учитывается не только G^{best} , но и лучший вариант по нескольким критериям из глобального архива частиц-лидеров.

В данном исследовании использована одна из модификаций роя частиц — OMOPSO [4], она отличается тем, что помимо использования архива лидеров, в ней применяется алгоритм CDA [22] для отбора лидеров из архива и операторы мутации дополнительно к обновлению скорости. Также в отличие от стандартного PSO и MOPSO в данном алгоритме коэффициенты когнитивного и социального обучения c_1 и c_2 в каждой итерации определяются случайным образом в диапазоне $[1.5, 2]$, что ускоряет глобальный поиск, вес инерции w также задается случайным образом в диапазоне $[0.1, 0.5]$.

SMPSO

Алгоритм SMPSO (Speed-constrained Multi-objective PSO) является развитием MOPSO [24]. Ключевая особенность заключается в том, что гиперпараметры задаются случайным образом и обновляются в каждой итерации. Однако, это приводит к большой разнице между скоростями частиц. Чтобы лучше контролировать скорость был предложен коэффициент сжатия:

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}, \quad (11)$$

$$\varphi = \begin{cases} c_1 + c_2, & c_1 + c_2 > 4 \\ 0, & c_1 + c_2 \leq 4 \end{cases}. \quad (12)$$

С учетом коэффициента сжатия скорость будет определяться как:

$$v_i(t+1) = \chi \left(wv_i(t) + c_1r_1(P_i^{best} - x_i) + c_2r_2(G^{best} - x_i) \right). \quad (13)$$

Помимо этого, авторы вводят дополнительный механизм, в котором накопленная скорость каждой переменной j (в каждой частице) ограничивается следующим образом [24]:

$$v_{i,j}(t) = \begin{cases} \Delta_j, & v_{i,j}(t) > \Delta_j \\ -\Delta_j, & v_{i,j}(t) \leq -\Delta_j \\ v_{i,j}(t), & \text{в остальных случаях} \end{cases}, \quad (14)$$

где $\Delta_j = 0.5(upper_{limit_j} - lower_{limit_j})$; $upper_{limit_j}$ — верхний предел переменной j ; $lower_{limit_j}$ — нижний предел переменной j .

MOEA/D

Последним рассматриваемым алгоритмом является MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition) - это эволюционный алгоритм, основанный на разложении задач многокритериальной оптимизации на несколько однокритериальных подзадач [25]. Алгоритм основан на следующих принципах: MOEA/D использует вектор весов для разбиения многокритериальной задачи на несколько подзадач. Для декомпозиции используется метод Чебышева [25]. Каждая подзадача взаимодействует с ограниченным набором соседних подзадач. Это создает локальную структуру, что позволяет алгоритму более эффективно исследовать пространство решений. Соседство может быть определено на основе пространственного расстояния между весами, что обеспечивает хорошее покрытие всего пространства решений. Далее применяются стандартные операторы мутации и кроссовера. При обнаружении нового решения, доминирующего над существующим, алгоритм обновляет текущее решение. После завершения заданного количества итераций алгоритм возвращает набор решений, представляющих собой множество Парето.

5. Тестирование алгоритмов

В результате было выбрано 5 оптимизационных алгоритмов для многокритериальной оптимизации – NSGA-II, NSGA-III, OMOPSO, SMPSO, MOEA/D. Необходимо определить, какой алгоритм будет наиболее эффективен. Каждый имеет свои гиперпараметры, которые влияют на скорость и точность решения в конкретной задаче [3]. Поэтому для объективного сравнения алгоритмов между собой необходимо решать задачу мета-оптимизации для каждого метода и для каждой тестовой задачи. Мета-оптимизация – использование одного оптимизационного алгоритма для настройки гиперпараметров другого алгоритма [3]. Так как в каждом алгоритме несколько гиперпараметров, то задача мета-оптимизации также будет являться многокритериальной.

Эффективный оптимизационный алгоритм должен затрачивать минимальное время на вычисление и иметь максимальную точность. Для оценки точности существует большое количество метрик. Наиболее часто используются метрики: гиперобъем HV (H-Value) и IGD (Inverted Generational Distance) [3, 12, 26].

IGD — это метрика, используемая для оценки качества решения в задачах многокритериальной оптимизации:

$$IGD(A) = \frac{1}{|Z|} \left(\sum_{i=1}^{|Z|} \hat{d}_i^p \right)^{\frac{1}{p}} \rightarrow \min, \quad (15)$$

где A – набор решений, Z – истинный фронт, \hat{d} - евклидово расстояние между ближайшими точками A и Z ($p=2$). Сначала определяется истинный Парето-фронт, который представлен как множество векторов, каждый из которых соответствует оптимальному решению. Затем для каждого решения, полученного в результате работы алгоритма, вычисляется минимальное расстояние до ближайшего решения в истинном фронте. Суммируются все рассчитанные расстояния. IGD позволяет оценить разнообразие и точность решений. Чем ближе значение IGD к 0, тем более разнообразное и точное решение получено.

HV — представляет собой метрику, оценивающую объем всех найденных решений:

$$|\Delta HV| \rightarrow \min. \quad (16)$$

Одновременно она позволяет оценить разнообразие полученных решений и качество охвата всего фронта-Парето, если известно значение HV для истинного фронта. В задаче сравнения оптимизационных алгоритмов необходимо использовать ΔHV , которая будет являться разницей между HV найденных решений и HV истинного фронта. Данная метрика позволяет оценить, совпадает ли найденный фронт Парето с истинным, не замыкается ли он

в локальном минимуме или как сильно отличаются найденные решения от истинных. Чем ближе значение $|\Delta HV|$ к нулю, тем точнее решение:

Для решения задачи мета-оптимизации используется более простой подход - метод взвешенных сумм (или алгоритм взвешенных целевых функций). Основная идея алгоритма заключается в том, что каждому критерию присваивается вес, который отражает его относительную важность по сравнению с другими критериями [3]. Затем для каждой альтернативы рассчитывается взвешенная сумма значений по всем критериям. Это можно выразить следующей формулой:

$$F(f(\bar{X})) = w_1 f_1(\bar{X}) + \dots + w_i f_i(\bar{X}), \quad (17)$$

где w – весовой коэффициент, f – целевая функция, F – взвешенная сумма.

В качестве целевых функций использовались IGD, HV и время расчета в секундах. Весовые коэффициенты для IGD и HV заданы равными 1, а весовой коэффициент для времени – 0.1:

$$F = IGD \cdot \frac{1}{\text{кг}} + |\Delta HV| \cdot \frac{1}{\text{кг}^3} + 0.1 \cdot t \cdot \frac{1}{\text{с}} \rightarrow \min. \quad (18)$$

Для решения задачи мета-оптимизации, которая была сведена к одноцелевой, был выбран алгоритм Байесовской оптимизации, так как он часто используется для настройки других алгоритмов [27]. Метод заключается в итеративном выборе новых конфигураций гиперпараметров, опираясь на текущую модель, а затем актуализации этой модели [27]. Такой подход позволяет эффективно собирать информацию о функции, и о её оптимальном значении. Данный подход был использован для проведения серии вычислительных экспериментов.

Описанные алгоритмы оптимизации и мета-оптимизации были реализованы на языке программирования Python, для соблюдения равных условий все алгоритмы запускались только на 1 ядре. На Рис. 2 приведены взвешенные суммы для каждого алгоритма и каждой тестовой задачи, в Таблице 3 приведены оптимальные гиперпараметры и целевые функции. Зеленым цветом выделены минимальные суммы для каждой задачи. Как видно, в задачах DTLZ2 и MW8 лидирует алгоритм NSGA-III, эти задачи являются наименее сложными и требуют меньших вычислительных ресурсов среди представленных задач. В более сложных DTLZ1, WFG1, WFG2 алгоритм SMPSO является наиболее эффективным. При этом алгоритм NSGA-II близок к NSGA-III, но наличие опорных направлений в последнем делает его более эффективным для задач с тремя целевыми функциями.

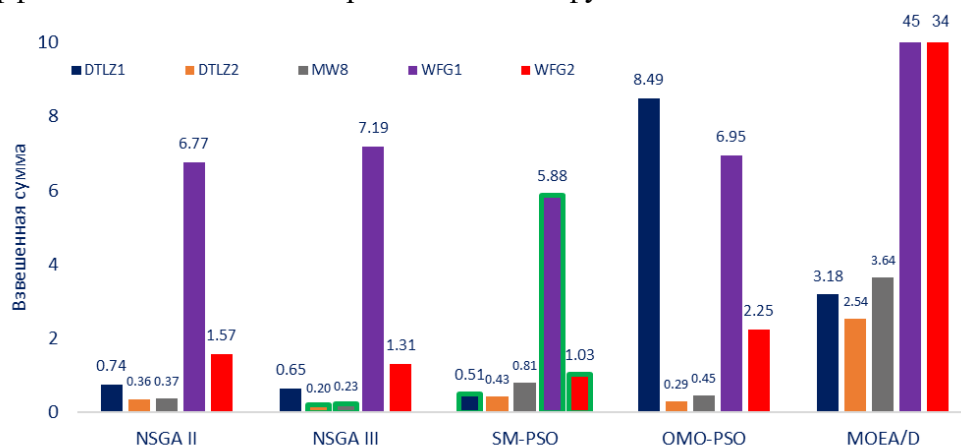


Рис. 2. Эффективность оптимизационных алгоритмов на тестовых функциях

Таб. 3. Результаты мета-оптимизации алгоритмов на тестовых задачах

Алгоритм	Параметры	DTLZ1	DTLZ2	MW8	WFG1	WFG2
NSGA II	Размер популяции	276	285	472	360	484
	Кол-во поколений	292	103	51	1719	245
	Вероятность кроссовера	0.97	0.8	0.99	0.89	0.82
	Коэффициент кроссовера	16	29	30	26	14
	Вероятность мутации	0.13	0.9	0.03	0.63	0.7
	Коэффициент мутации	7	29	30	12	19
	Взвешенная сумма	0.74	0.36	0.37	6.77	1.57
NSGA III	Размер популяции	55	52	50	298	198
	Кол-во поколений	321	101	157	295	280
	Вероятность кроссовера	0.95	0.98	0.99	0.05	0.92
	Коэффициент кроссовера	30	2	30	30	28
	Вероятность мутации	0.57	0.96	0.41	0.83	0.56
	Коэффициент мутации	6	30	4	3	3
	Кол-во опорных направлений	28	11	10	5	18
	Взвешенная сумма	0.65	0.2	0.23	7.19	1.31
SMPSO	Размер роя	54	56	59	230	192
	Кол-во итераций	11340	4592	11564	2300	2496
	Вероятность мутации	0.02	0.05	0.12	0.02	0.03
	Взвешенная сумма	0.51	0.43	0.81	5.88	1.03
OMOPSO	Размер роя	55	33	46	30	200
	Кол-во итераций	83734	1566	3751	7337	14098
	Эпсилон	0.3	0.2	0.11	0.03	0.05
	Размер архива лидеров	300	293	44	35	31
	Вероятность мутации	0.05	0.88	0.36	0.96	0.99
	Взвешенная сумма	8.49	0.29	0.45	6.95	2.25
MOEA/D	Кол-во соседей у особи	261	209	368	487	495
	Кол-во поколений	37	13	6	3	2
	Вероятность скрещивания соседей	0.57	0.8	0.66	0.97	0.95
	Вероятность кроссовера	0.58	0.99	0.62	0.69	0.9
	Коэффициент кроссовера	25	30	7	29	28
	Вероятность мутации	0.36	0.56	0.54	0.96	0.98
	Коэффициент мутации	12	30	20	4	3
	Взвешенная сумма	3.18	2.54	3.64	45.28	34.29

По полученным данным можно сделать вывод о том, что алгоритмы NSGA-III и SMPSO наиболее эффективны для данных задач, при этом SMPSO лучше работает с задачами с большей вычислительной сложностью, а NSGA-III – с более простыми. Такое разделение связано с тем, что NSGA-III предназначен для поиска локальных решений и поддерживает разнообразие решений (минимальное IGD), а SMPSO лучше показывает себя в глобальной оптимизации [12].

6. Гибридизация алгоритмов

Для быстрого выбора свободных параметров моделей фильтрации необходимо, чтобы алгоритм был эффективен во всех типах тестовых функций, так как форма Парето-фронта может быть разной в зависимости от эксперимента и свойств горной породы. Предлагается совместить алгоритм NSGA-III и SMPSO, чтобы использовать их сильные стороны (локальный и глобальный поиск) в одном алгоритме. Согласно классификации гибридных алгоритмов [3] одним из способов объединить два различных алгоритма является гибридизация типа постпроцессор/препроцессор. Идея заключается в том, чтобы применять алгоритмы последовательно. Можно комбинировать эффективный алгоритм глобального поиска с последовательно подключаемым алгоритмом локального поиска (Рис. 3, 1-2). Такой подход уже был реализован, например, в работе [11] сначала применялся генетический алгоритм GA, затем рой частиц PSO, такой гибрид оказался эффективнее исходных алгоритмов.

Реализовано два последовательных гибридных алгоритма типа процессор/постпроцессор. В первом сначала запускается алгоритм SMPSO и затем NSGA-III, во втором – наоборот. Проведена аналогичная с предыдущими рассмотренными алгоритмами мета-оптимизация. Результаты представлены на Рис. 4. Как видно, вариант с последовательной работой NSGA-III - SMPSO показал себя значительно хуже, чем отдельные алгоритмы. Обратная последовательность SMPSO – NSGA-III, наоборот, в четырех задачах из пяти показала

большую эффективность. Связано это с тем, что алгоритмы на основе роя частиц быстрее находят истинный Парето-фронт, но на равномерное распределение частиц роя уходит много времени. Генетические алгоритмы же, напротив, долго приближаются к истинному фронту, но при его обнаружении быстро получают равномерное распределение.



Рис. 3. Варианты гибридизации алгоритмов

1, 2 – тип процессор/постпроцессор, 3 – параллельный гибрид

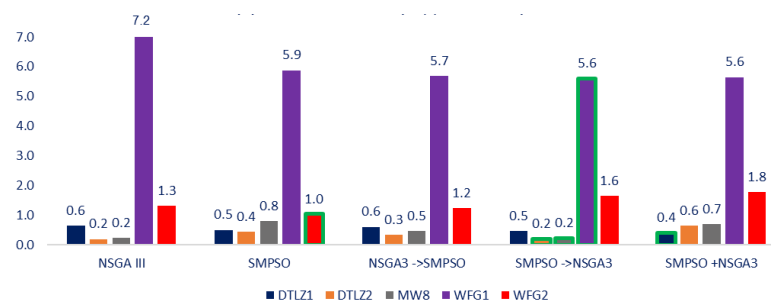


Рис. 4. Результаты мета-оптимизации гибридных алгоритмов в сравнении с моно-алгоритмами

Последовательный SMPSO-NSGA-III показал высокую эффективность, но в задаче WFG2 проявил себя хуже, чем обычный SMPSO. Возможным решением является применение гибридизации другого вида. Предлагается запускать два алгоритма параллельно, а в процессе производить обмен лучшими решениями между алгоритмами. Существует множество вариаций таких гибридов. Так, в работах [12, 13] популяции NSGA-III и MOPSO смешиваются случайным образом. В работе [14] предлагается новый гибрид SONG из NSGA-II и SMPSO, где после каждой итерации роя частиц производится отбор частиц на основе алгоритма greedy selection (жадный отбор) для передачи их в NSGA-II. Во всех этих исследованиях гибридные алгоритмы показывали более высокую производительность, чем исходные одиночные алгоритмы. Поэтому в данной работе протестирован гибридный алгоритм SMPSO+NSGA-III, в котором алгоритмы запускаются параллельно, а после каждой итерации популяции случайным образом перемешиваются между собой (Рис. 3, 3). Результаты мета-оптимизации приведены на Рис. 4. Как видно, такой подход дает прирост в эффективности только для задачи DTLZ1, в остальных задачах он проигрывает другим алгоритмам.

Тестирование последовательного гибрида SMPSO-NSGA-III показало, что ключевой проблемой, влияющей на эффективность, является выборка решений при переходе от SMPSO к NSGA-III. Популяция представлена различными группами частиц (Рис. 5, А и В), некоторые группы находятся близко к истинному фронту Парето, другие кластера являются вылетами и находятся далеко. Если такие кластера передавать в NSGA-III, то качество и скорость оптимизации снижаются. Например, на Рис. 5, А выделяется облако точек, которое уже находится внутри истинного Парето-фронта, а несколько точек находятся за его пределами, передавать эти точки в качестве начальной популяции в NSGA-III увеличит время его работы. На Рис. 5, В частицы отдалены от истинного фронта намного сильнее. Необходимо отделять такие частицы при переносе популяции из одного оптимизационного алгоритма в другой. Данная логика реализована в работе [28], где предлагается гибрид PSO и

GA, который делится на несколько отдельных роев, при этом лучшие по целевой функции решения роя частиц передаются в генетический алгоритм. Выбирать лучшие решения по абсолютному значению целевых функций является простым, но не наиболее эффективным решением, так как в случае Рис. 5, А достаточно отсечь только две точки, а в варианте Рис. 5, В необходимо отсечь 12 точек. Если выбирать по значению целевых функций, то необходимо знать, сколько частиц нужно отделить от популяции, а эта информация заранее неизвестна.

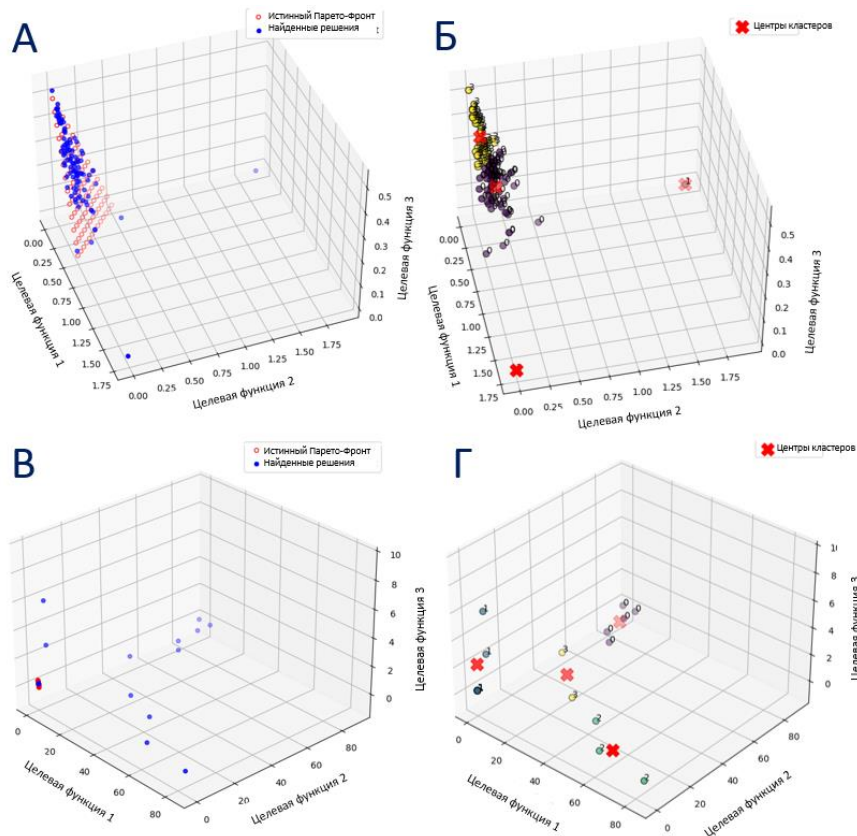


Рис. 5. Промежуточные результаты расчета SMPSO для задачи DTLZ1

А – 200 итераций, Б – кластеризация для 200 итераций, В – 100 итераций, Г – кластеризация для 100 итераций

Возможным решением является использование методов кластеризации для выделения наиболее перспективных наборов частиц. Наиболее известным методом кластеризации является алгоритм К-средних (K-means) [29]. Это один из самых простых методов кластеризации, который используется для разделения набора данных на заданное количество кластеров k . Его простота позволяет производить кластеризацию с небольшими вычислительными затратами, что существенно не замедлит работу оптимизационного алгоритма. Ранее алгоритм К-средних использовался для кластеризации опорных направлений в алгоритме NSGA-III [30]. Вместо всего набора опорных направлений генетическому алгоритму подавались только центры кластеров, алгоритм назвали NSGA-III-GKM, а результаты моделирования показали, что алгоритм демонстрирует лучшие производительность, разнообразие и сходимость. Применительно к гибриднему алгоритму предлагается использовать кластеризацию К-средних в конечной популяции SMPSO для выборки наиболее перспективных кластеров для передачи их в NSGA-III. Например, в варианте Рис. 5, А кластеризация приведена на Рис. 5, Б, видно, что два кластера лежат внутри истинного фронта, а дальние точки выделяются в два других кластера. В варианте Рис. 5, В кластеризация приведена на Рис. 5, Г, в этом случае выделяется один кластер близкий к фронту Парето, а три остальных находятся на большом расстоянии. Количество

кластеров в данных вариантах задано равным четырем, однако, этот параметр необходимо варьировать в процессе мета-оптимизации. Так как количество кластеров заранее неизвестно, предлагается рассчитывать расстояние от центра кластеров до идеальной точки (0, 0, 0), затем половину кластеров с наименьшими расстояниями передавать в NSGA-III. Алгоритм с кластеризацией приведен на Рис. 6. Результаты мета-оптимизации данного подхода представлены на Рис. 7. Данный алгоритм значительно эффективнее в задачах DTLZ1, DTLZ2 и сравним по производительности в задачах MW8 и WFG1, а в задаче WFG2 он по-прежнему проигрывает обычному SMPSO.



Рис. 6. Гибридный алгоритм SMPSO и NSGA-III с кластеризацией K-means

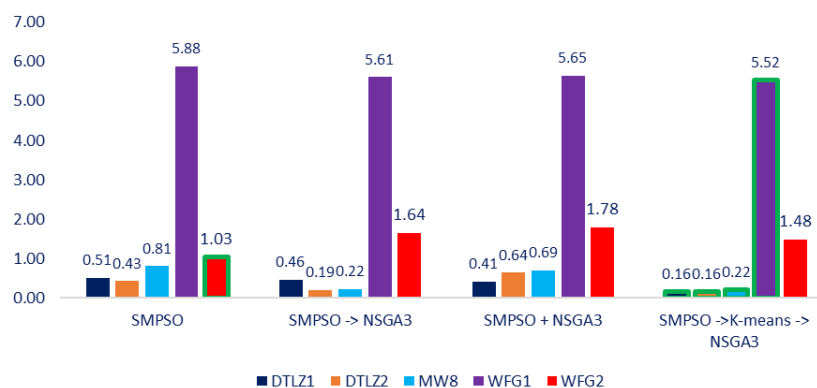


Рис. 7. Результаты мета-оптимизации гибридного алгоритма с кластеризацией в сравнении с другими алгоритмами

Чтобы получить алгоритм, который будет наиболее эффективным сразу во всех тестовых задачах был произведен анализ изменения качества решения во времени. Построены зависимости относительного изменения метрик в процентах от номера итерации, результаты представлены на Рис. 8. Было определено, что в алгоритме SMPSO при решении тестовых задач значение HV сначала быстро изменяется, а затем алгоритм находит истинный фронт Парето и изменение метрики замедляется (Рис. 8, А). Итерации с небольшим изменением HV отрицательно сказываются на эффективности алгоритма, так как значительно не улучшают решение, но тратят расчетное время.

На Рис.8, Б приведена аналогичная динамика изменения IGD для NSGA-III. Видно, что в начале IGD быстро меняется – ведется поиск решений, а затем производятся ненужные итерации. Такой же эффект был показан в работе [13]. Авторами была построена аналогичная форма зависимости метрики от номера итерации. Подобная динамика изменения точности решения свойственна для многих методов оптимизации, так как когда решение найдено, все последующие итерации только увеличивают время работы алгоритма, но не его точность. Избежать этого позволяют критерии остановки алгоритма.

В предлагаемом гибридном алгоритме предлагается ввести внутренний алгоритм проверки динамики изменения метрик HV и IGD с целью остановки алгоритма при незначительном их изменении. Однако, использовать IGD не получится, так как в реальной задаче истинный фронт Парето неизвестен, а для расчета данной метрики он необходим, поэтому вместо него предлагается использование другого параметра, отвечающего за разнообразие решение – Spacing (SP). Данная метрика представляет собой среднее евклидово

расстояние между точками найденного решения, а не между точками решения и истинного фронта, как в случае IGD [31].

$$SP = \sqrt{\frac{1}{|Z| - 1} \sum_{i=1}^{|Z|} (d_i - \bar{d})^2}, \quad (19)$$

где d_i - минимальное Манхэттенское расстояние между решением i и всеми остальными, \bar{d} - среднее этих величин. Динамика изменения SP приведена на Рис. 8, В, где наблюдается аналогичное затухание изменений метрики.

В результате реализована логика: если на протяжении n поколений метрика меняется менее чем на $y\%$, то алгоритм останавливается, блок-схема приведена на Рис. 9. Для определения необходимых критериев выхода из алгоритмов была проведена Байесовская мета-оптимизация, в которой гиперпараметры алгоритмов были фиксированы (100 частиц, 100 поколений и т.п.), а варьировались только предельный процент отклонения y и количество поколений проверки n для выхода из алгоритмов. Результаты представлены на в Таблице 4. Исходя из полученных данных, предлагается использовать средние параметры остановки алгоритмов: останавливать SMPSO, если HV меняется менее чем на 3% за 10 итераций и останавливать NSGA-III, если SP меняется менее чем на 3% за 18 итераций. Осреднение параметров было использовано с целью обеспечения улучшения производительности алгоритма для всех типов задач.

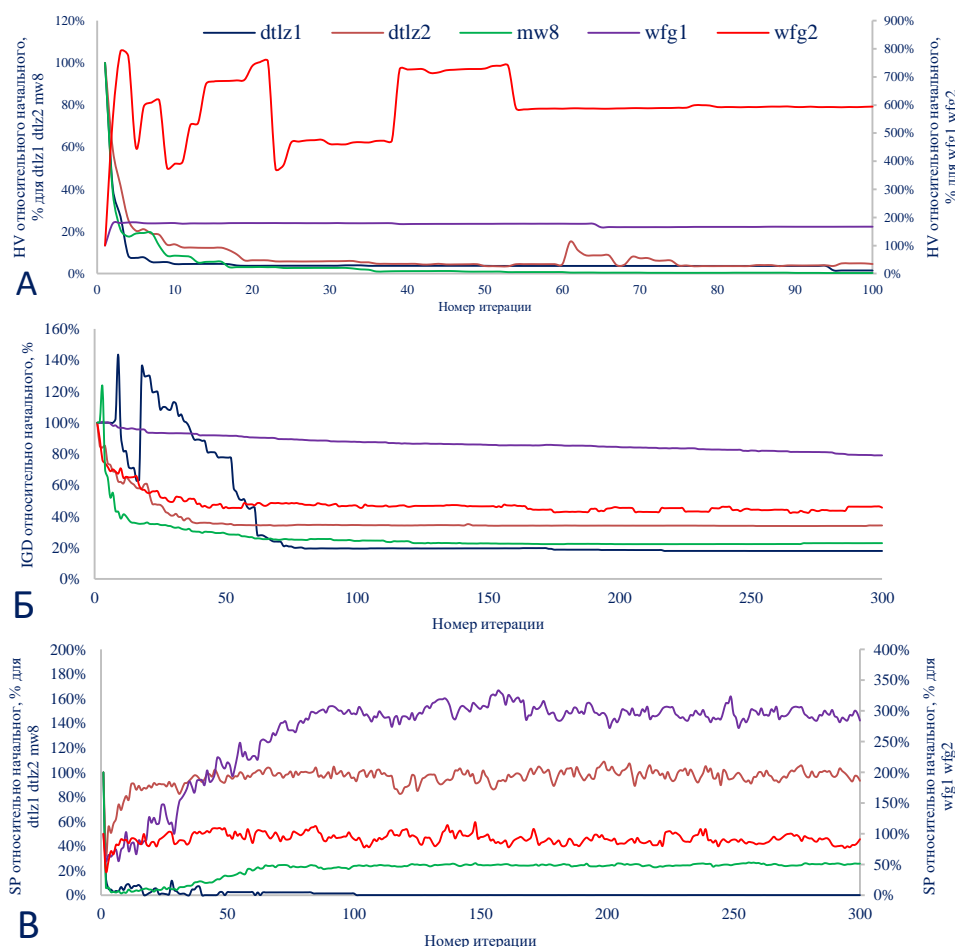


Рис. 8. Динамика изменения метрик для различных тестовых задач

А – Динамика изменения Н-value для SMPSO, Б – Динамика изменения IGD для NSGA-III, В – Динамика изменения SP для NSGA-III

В результате был получен новый гибридный алгоритм SMPSO-NSGA-III типа процессор/постпроцессор с применением алгоритма кластеризации K-means и оптимизированными критериями выхода на основе динамики HV и SP. На Рис. 10, А приведено сравнение эффективности полученного алгоритма с ранее рассмотренными, по всех тестовых функциях новый алгоритм превосходит остальные по эффективности. На Рис. 10, Б приведены приросты эффективности, так в задаче DTLZ1 прирост составил 311%, когда как в задаче WFG2 – 7%.



Рис. 9. Итоговый гибридный алгоритм оптимизации SMPSO и NSGA-III с кластеризацией K-means и динамическими критериями остановки

Таб. 4. Результаты мета-оптимизации параметров остановки алгоритмов

Параметр		Среднее	DTLZ1	DTLZ2	MW8	WFG1	WFG2
NSGA-III SP	Минимальное отклонение, %	3	4.29	2.9	2.4	1.8	1.5
	Кол-во точек для проверки	18	20	11	22	12	25
SMPSO HV	Минимальное отклонение, %	3	2.23	2.95	2.88	1.95	2.81
	Кол-во точек для проверки	10	16	5	8	8	15

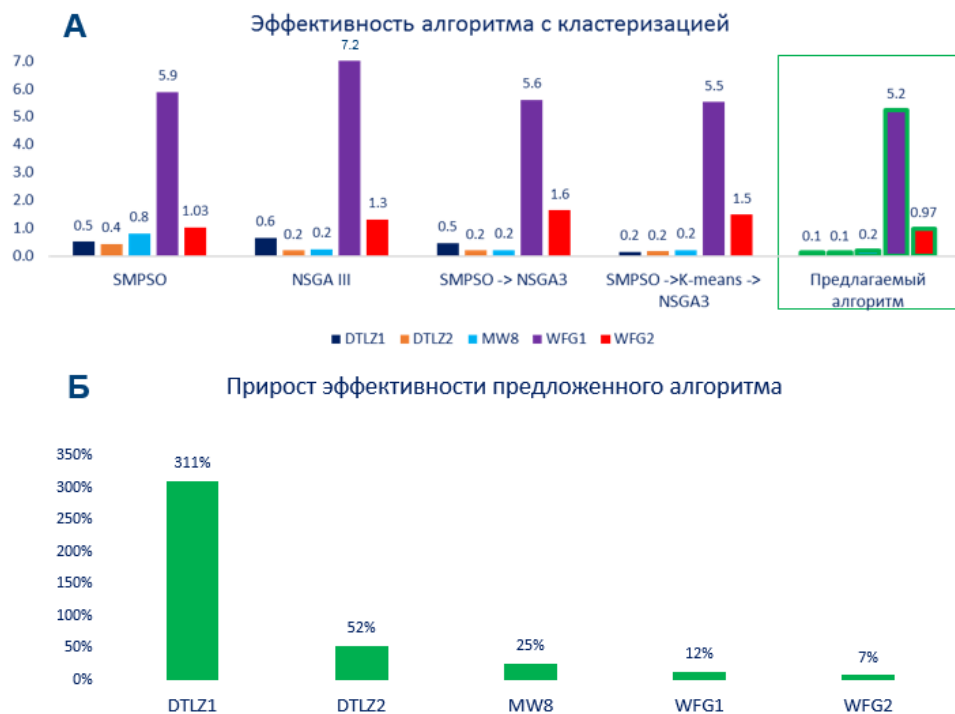


Рис. 10. Сравнение итогового алгоритма с моно-алгоритмами и более простыми гибридами
А – сравнение по взвешенной сумме, Б – прирост эффективности относительно моно-алгоритмов

7. Тестирование предлагаемого алгоритма на реальной задаче

Разработанный гибридный оптимизационный алгоритм необходимо применить на исходной задаче восстановления свободных параметров модели капиллярной пропитки для воспроизведения экспериментов. Для тестирования использованы данные из исследования Танга и Фирузабади [16]. Выбран эксперимент по насыщению керна с давлениями закачки 0, 0.33 и 4.65 атм. Был использован разработанный алгоритм оптимизации, тремя целевыми функциями являлись сумма модулей невязок между накопленной массой пропитки в модели и в эксперименте в каждой временной точке для трех экспериментов (8). Из построенного Парето-фронта лучшее решение выбиралось как наиболее близкое к идеальной точке (0,0,0). Гиперпараметры разработанного алгоритма определялись как средние по оптимальным параметрам мета-оптимизации тестовых задач, они приведены в Таблице 5.

Таб. 5. Оптимальные гиперпараметры разработанного гибридного алгоритма							
Алгоритм	Оптимальные гиперпараметры						Среднее
SMPSO	Размер роя	80	66	137	163	110	111
	Кол-во итераций	1603	39	33	2197	337	842
	Вероятность мутации	0.25	0.29	0.18	0.39	0.63	0.35
	Вероятность кроссовера	0.96	0.98	0.96	0.05	0.08	0.61
	Коэффициент кроссовера	9	23	24	4	18	16
NSGA-III	Вероятность мутации	0.64	0.98	0.46	0.88	0.29	0.65
	Коэффициент мутации	14	30	3	17	2	13
	Кол-во опорных направлений	13	35	45	35	44	34
K-Means	Количество кластеров	4	5	6	5	5	5

Фактические зависимости накопленного объема закачанной воды от времени приведены на Рис. 11, также приведены результаты ручной настройки свободных параметров модели на результаты эксперименты из предыдущей работы [2] вместе с результатами применения гибридного алгоритма. Как видно, предложенный гибридный алгоритм позволяет точно воспроизвести все эксперименты, оптимальное решение лежало в координатах невязок с экспериментом - (0.0014, 0.0050, 0.0012). Решение заняло 95 итераций/поколений и 275 секунд.

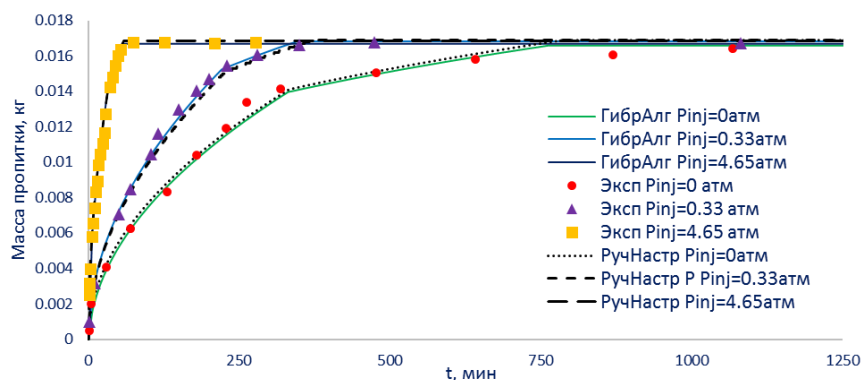


Рис. 11. Сравнение ручного подбора параметров с гибридным алгоритмом

Также для подтверждения эффективности разработанного алгоритма были проведены расчеты алгоритмов SMPSO и NSGA-III с осредненными оптимальными гиперпараметрами из результатов мета-оптимизации и с заданным количеством итераций, равным итерациям в гибридном алгоритме. Результаты приведены на Рис. 12. Алгоритм SMPSO просчитал 95 итераций за 271 секунду, оптимальная точка имеет координаты (0.0186, 0.0185, 0.0058). Алгоритм NSGA-III отработал 95 итераций за 282 секунды, оптимальная точка имеет

координаты (0.0018, 0.0068, 0.0749). По координатам оптимальных точек трех алгоритмов видно, что все целевые функции (невязки с экспериментами) предложенного гибридного метода меньше, чем невязки в оригинальных алгоритмах. Снижение невязок в предлагаемом алгоритме составило от 22% до 79%.

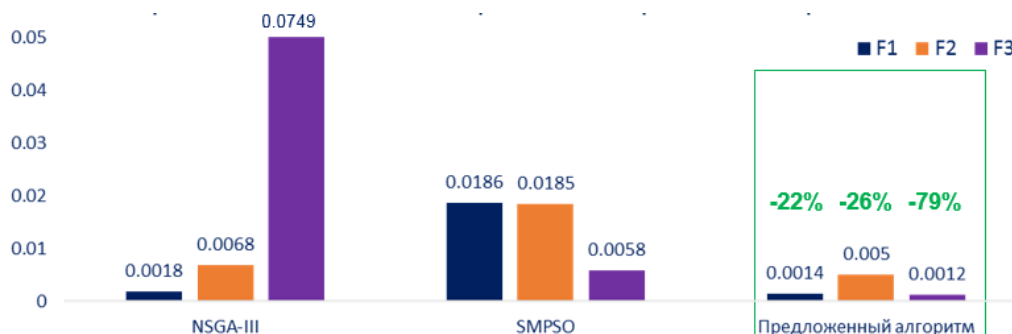


Рис. 12. Сравнение оптимального решения гибридного алгоритма с исходными алгоритмами

Заключение

В рамках данного исследования было протестировано пять мета-эвристических алгоритмов многокритериальной оптимизации на пяти тестовых функциях. Наиболее эффективными по соотношению точности и скорости решения стали роевой алгоритм SMPSO и генетический NSGA-III. С целью получения более универсального алгоритма был создан гибридный алгоритм типа процессор/постпроцессор, для улучшения точности и производительности был применен метод кластеризации для отбора лучших решений для передачи из одного алгоритма в другой. Также были разработаны динамические критерии останова алгоритмов на основе метрик H-value и Spacing. Полученный гибридный алгоритм был использован для оптимизации свободных параметров модели капиллярной пропитки для настройки на экспериментальные данные. Разработанный метод показал большую эффективность и лучшую сходимость с фактическими данными по сравнению с оригинальными алгоритмами SMPSO и NSGA-III. За то же самое расчетное время алгоритм получил меньшие невязки между моделью и тремя экспериментами на 22%, 26% и 79% соответственно. Разработанный алгоритм может применяться для похожих обратных задач, где необходимо восстановить параметры фильтрационной модели по результатам серии из нескольких экспериментов.

Литература

- [1] Методические рекомендации по определению подсчетных параметров залежей нефти и газа по материалам геофизических исследований скважин с привлечением результатов анализов керна, опробований и испытаний продуктивных пластов / под редакцией Б. Ю. Вендельштейна, В. Ф. Козяра, Г. Г. Яценко. — Калинин: НПО «Союзпромгеофизика», 1990. — 261 с
- [2] Цыкунов О.И. Математическая модель фильтрации с учетом капиллярных сил в мультимедийной среде / О. И. Цыкунов // International Journal of Open Information Technologies. – 2024. – Т. 12, № 10. – С. 45-55. – EDN NGFAIN.
- [3] Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой / А.П. Карпенко. - Москва : МГТУ им. Н.Э. Баумана, 2017. - 446 с.
- [4] Sierra, M. R., & Coello Coello, C. A. (2005). Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and ϵ -Dominance. In Lecture Notes in Computer Science (pp. 505–519). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-31880-4_35

- [5] Kennedy, J., & Eberhart, R. (n.d.). Particle swarm optimization. In Proceedings of ICNN'95 - International Conference on Neural Networks (Vol. 4, pp. 1942–1948). ICNN'95 - International Conference on Neural Networks. IEEE.
<https://doi.org/10.1109/icnn.1995.488968>
- [6] 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. In IEEE Transactions on Evolutionary Computation (Vol. 18, Issue 4, pp. 577–601). Institute of Electrical and Electronics Engineers (IEEE).
<https://doi.org/10.1109/tevc.2013.2281535>
- [7] Jain, H., & Deb, K. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. In IEEE Transactions on Evolutionary Computation (Vol. 18, Issue 4, pp. 602–622). Institute of Electrical and Electronics Engineers (IEEE).
<https://doi.org/10.1109/tevc.2013.2281534>
- [8] Blank, J., & Deb, K. (2020). Pymoo: Multi-Objective Optimization in Python. In IEEE Access (Vol. 8, pp. 89497–89509). Institute of Electrical and Electronics Engineers (IEEE).
<https://doi.org/10.1109/access.2020.2990567>
- [9] Гарагулова Анастасия Керимовна, Горбачева Дарья Олеговна, & Чирков Денис Владимирович (2018). Сравнение генетических алгоритмов MOGA и NSGA-II на задаче оптимизации формы рабочего колеса гидротурбины. Вычислительные технологии, 23 (5), 21-36.
- [10] Indraneel Das and J. E. Dennis. Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM J. on Optimization, 8(3):631–657, March 1998. URL: <http://dx.doi.org/10.1137/S1052623496307510>
- [11] Cherki, I., Chaker, A., Djidar, Z., Khalfallah, N., & Benzergua, F. (2019). A Sequential Hybridization of Genetic Algorithm and Particle Swarm Optimization for the Optimal Reactive Power Flow. In Sustainability (Vol. 11, Issue 14, p. 3862). MDPI AG.
<https://doi.org/10.3390/su11143862>
- [12] Truong, N. H., & Dao, D.-N. (2020). New hybrid between NSGA-III with multi-objective particle swarm optimization to multi-objective robust optimization design for Powertrain mount system of electric vehicles. In Advances in Mechanical Engineering (Vol. 12, Issue 2, p. 168781402090425). SAGE Publications. <https://doi.org/10.1177/1687814020904253>
- [13] Aboura, F. (2019). Tuning PID Controller Using Hybrid Genetic Algorithm Particle Swarm Optimization Method for AVR System. In 2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM).
<https://doi.org/10.1109/acemp-optim44294.2019.9007124>
- [14] Hussain, Md. M., Azar, A. T., Ahmed, R., Umar Amin, S., Qureshi, B., Dinesh Reddy, V., Alam, I., & Khan, Z. I. (2023). SONG: A Multi-Objective Evolutionary Algorithm for Delay and Energy Aware Facility Location in Vehicular Fog Networks. In Sensors (Vol. 23, Issue 2, p. 667). MDPI AG. <https://doi.org/10.3390/s23020667>
- [15] Cai, J., Perfect, E., Cheng, C.-L., & Hu, X. (2014). Generalized Modeling of Spontaneous Imbibition Based on Hagen–Poiseuille Flow in Tortuous Capillaries with Variably Shaped Apertures. In Langmuir (Vol. 30, Issue 18, pp. 5142–5151). American Chemical Society (ACS). <https://doi.org/10.1021/la5007204>
- [16] Tang, G.-Q., & Firoozabadi, A. (2001). Effect of Pressure Gradient and Initial Water Saturation on Water Injection in Water-Wet and Mixed-Wet Fractured Porous Media. In SPE Reservoir Evaluation & Engineering (Vol. 4, Issue 06, pp. 516–524). Society of Petroleum Engineers (SPE). <https://doi.org/10.2118/74711-pa>
- [17] O'Mahony, C., & Wilson, N. (2012). Sorted Pareto Dominance: An Extension to Pareto Dominance and Its Application in Soft Constraints. In 2012 IEEE 24th International Conference on Tools with Artificial Intelligence (Vol. 1996, pp. 798–805). 2012 IEEE 24th

- International Conference on Tools with Artificial Intelligence (ICTAI 2012). IEEE. <https://doi.org/10.1109/ictai.2012.113>
- [18] McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. In *Technometrics* (Vol. 21, Issue 2, pp. 239–245). Informa UK Limited. <https://doi.org/10.1080/00401706.1979.10489755>
- [19] Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2002). "Scalable multi-objective optimization test problems." In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)* (pp. 825-830). IEEE.
- [20] Ma, M., & Wang, L. (2018). "A new set of benchmark multi-objective optimization problems with complicated Pareto sets and Pareto fronts of arbitrary shapes." In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '18)* (pp. 791-798). ACM.
- [21] Huband, S., Hingston, P., Barone, L., & While, L. (2006). "A review of multiobjective test problems and a scalable test problem toolkit." *IEEE Transactions on Evolutionary Computation*, 10(5), 477-506.
- [22] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. In *IEEE Transactions on Evolutionary Computation* (Vol. 6, Issue 2, pp. 182–197). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/4235.996017>
- [23] Deb, K. *Multi-objective optimization using evolutionary algorithms*. New York: John Wiley & Sons, 2001. 497 p.
- [24] Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello Coello, C. A., Luna, F., & Alba, E. (2009). SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making(MCDM)*. 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM). IEEE. <https://doi.org/10.1109/mcdm.2009.4938830>
- [25] Qingfu Zhang, & Hui Li. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. In *IEEE Transactions on Evolutionary Computation* (Vol. 11, Issue 6, pp. 712–731). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/tevc.2007.892759>
- [26] Jeong, S., Obayashi, S., & Minemura, Y. (2008). Application of hybrid evolutionary algorithms to low exhaust emission diesel engine design. In *Engineering Optimization* (Vol. 40, Issue 1, pp. 1–16). Informa UK Limited. <https://doi.org/10.1080/03052150701561155>
- [27] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms (Version 2). arXiv. <https://doi.org/10.48550/ARXIV.1206.2944>
- [28] Shi, X. H., Lu, Y. H., Zhou, C. G., Lee, H. P., Lin, W. Z., & Liang, Y. C. (n.d.). Hybrid evolutionary algorithms based on PSO and GA. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.* (Vol. 4, pp. 2393–2399). The 2003 Congress on Evolutionary Computation, 2003. CEC '03. IEEE. <https://doi.org/10.1109/cec.2003.1299387>
- [29] Steinley, Douglas. (2006). K-means clustering: A half-century synthesis. In *British Journal of Mathematical and Statistical Psychology* (Vol. 59, Issue 1, pp. 1–34). Wiley. <https://doi.org/10.1348/000711005x48266>
- [30] Liu, Q., Liu, X., Wu, J., & Li, Y. (2019). An Improved NSGA-III Algorithm Using Genetic K-Means Clustering Algorithm. In *IEEE Access* (Vol. 7, pp. 185239–185249). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/access.2019.2960531>
- [31] Zheng, K., Yang, R., Xu, H., & Hu, J. (2017). A new distribution metric for comparing Pareto optimal solutions. *Structural and Multidisciplinary Optimization*, 55, 53-62. <https://doi.org/10.1007/S00158-016-1469-3>.

Development of a hybrid algorithm for multi-criteria optimization for restoring the properties of a porous medium from filtration experiments

Tsykunov O.I.^{1,2,*}

¹Gazpromneft STC, Saint-Petersburg, Russian Federation

²St. Petersburg Polytechnic University of Peter the Great, Saint-Petersburg, Russian Federation³

*olegtsykunovmgdn@gmail.com

Abstract. The problem of determining the physical properties of a porous medium based on filtration experiments is considered. A two-component model of capillary imbibition in a medium with double porosity is used in the form of a system of three differential equations, which contains 10 free parameters. Optimization algorithms are used to automate their determination based on a series of three experiments. The model's compliance with each experiment is an objective function, so the problem of multicriteria optimization is considered. The following metaheuristic algorithms were selected for testing: NSGA-II, NSGA-III, SMPSO, OMOPSO, MOEA/D. For each algorithm, Bayesian meta-optimization of hyperparameters was performed for 5 test functions: DTLZ1, DTLZ2, MW8, WFG1, WFG2. It was determined that the most effective algorithms in terms of accuracy and computational time are SMPSO and NSGA-III. SMPSO is more efficient in problems with greater computational complexity, and NSGA-III is more efficient in simple problems. To obtain an efficient optimization method in all test functions, a hybrid algorithm was created. Various hybridization methods were tested, the best option was processor/postprocessor hybridization, where SMPSO is initiated first, after which the population is transferred to NSGA-III. To improve the hybrid algorithm, the K-means clustering method was used to select the most promising particle clusters from the SMPSO population in order to transfer them to NSGA-III. An adaptive algorithm stopping criterion was also introduced based on tracking the dynamics of changes in the metrics of the obtained solutions - H-value and Spacing. Comparison of the obtained algorithm with swarm and genetic algorithms showed that the proposed algorithm finds a combination of free parameters with a smaller discrepancy between the model and the experiment for the same number of iterations.

Keywords: multicriteria optimization, particle swarm optimization, genetic algorithm, meta-optimization, capillary imbibition model, NSGA-III, SMPSO