

Модифицированный метод адаптивных взвешенных сумм в задаче многокритериальной оптимизации

11, ноябрь 2013

DOI: 10.7463/1113.0632468

Карпенко А. П., Савелов А. С.

УДК 519.6

Россия, МГТУ им. Н.Э. Баумана

apkarpenko@mail.ru

s_savelov@mail.ru

Введение

Постановка задачи многокритериальной оптимизации (МКО-задачи) фиксирует множество допустимых значений вектора варьируемых параметров задачи и вектор критериальных функций. Данная информация позволяет обычно выделить не одно решение задачи, но множество таких решений (множество Парето). Поэтому полагаем, как это часто делается в современных публикациях в данной области, что решением МКО-задачи является ее множество Парето. Множество Парето занимает в теории многокритериальной оптимизации исключительное место. Это, прежде всего, связано с тем, что согласно известному принципу Эджворта-Парето, при «разумном» поведении лица, принимающего решения (ЛПР), выбор решения следует производить на этом множестве [1].

Известно большое число *методов Парето-аппроксимации*, то есть методов построения некоторой конечномерной аппроксимации множества, а тем самым, и фронта Парето. Обзор таких методов представлен, например, в работе [2].

Решение задачи Парето-аппроксимации методом адаптивных взвешенных сумм (*Adaptive Weighted Sum (AWS) method*) предложили Рю, Ким и Ван (*J.-H. Ryu, S. Kim, H. Wan*) [3]. Метод основан на аддитивной

свертке частных критериев оптимальности, но, в отличие от классического метода суммы взвешенных критериев (*Weighted Sum (WS) method*) [1], также использующего такую свертку, предполагает адаптацию весовых коэффициентов в процессе итераций на основе информации о текущем положении подобласти поиска. Целью разработки метода AWS было преодоление известного недостатка метода WS, заключающегося в невозможности локализации точек множества Парето, которые соответствуют вогнутым фрагментам фронта Парето.

Для сокращения затрат на вычисление значений критериальных функций метод AWS использует метамоделли этих функций. Поскольку в практически значимых задачах вычислительная сложность критериальных функций обычно является высокой, это обстоятельство позволяет сократить вычислительные затраты на решение МКО-задачи и делает метод весьма перспективным, например, в задачах автоматизированного проектирования. Недостатком метода AWS в существующем виде является его ориентация на решение только двухкритериальных задач Парето-аппроксимации.

Авторы метода AWS выполнили исследование его эффективности для двух относительно простых тестовых задач многокритериальной оптимизации [3], которое показало высокую эффективность метода. В работе [4] мы повторили указанное исследование, а также выполнили аналогичное исследование для трех других сложных тестовых задач. Результаты этого исследования показали, что метод AWS дает высокое качество Парето-аппроксимации в случае выпуклого, хотя, быть может, и несвязного фронта Парето. Для задач, имеющих вогнутый фронт Парето, метод не всегда обеспечивает удовлетворительное качество решения или обеспечивает его, но при значительном числе итераций. В некоторых случаях метод дает недопустимые решения, обусловленные используемым способом учета ограничений на текущую подобласть поиска.

Работа открывает цикл из двух публикаций и имеет целью преодоление указанных недостатков метода AWS. Вторая работа будет посвящена

приложению представляемого в данной работе методического, алгоритмического и программного обеспечения к решению некоторых задач параметрической идентификации кинетических моделей реакций нейтрального металлокомплексного катализа.

В первом разделе работы даем постановку МКО-задачи. Второй раздел представляет базовые методы решения этой задачи. В третьем разделе рассматриваем несколько предложенных нами модификаций метода AWS. В четвертом разделе приводим краткое описание разработанного программного обеспечения, которое реализует метод AWS и его модификации. Пятый раздел посвящен исследованию эффективности указанных модификаций метода AWS. В заключении формулируем основные результаты работы и обсуждаем перспективы ее развития.

1. Постановка задачи

Пусть множеством допустимых значений вектора варьируемых параметров X является ограниченное и замкнутое множество $D_X \subset R^{|X|}$. Положим, что критериальная вектор-функция $F(X) = (f_1(X), f_2(X), \dots, f_{|F|}(X))$ со значениями в критериальном пространстве $R^{|F|}$ определена в области D_X . ЛПР стремится минимизировать в этой области каждый из частных критериев оптимальности $f_1(X), f_2(X), \dots, f_{|F|}(X)$, что условно записываем в виде

$$\min_{X \in D_X} F(X) = F(X^*) = F^*, \quad (1)$$

где векторы X^*, F^* - искомое решение задачи многокритериальной оптимизации. Здесь и далее запись вида $|A|$, где A - вектор или некоторое счетное множество, означает размерность этого вектора или мощность множества соответственно.

Вектор-функция $F(X)$ выполняет отображение множества D_X во множество $D_F \in R^{|F|}$, которое называется множеством достижимости.

Множество и фронт Парето задачи (1) обозначаем D_X^* , D_F^* соответственно;
 $D_X^* \subseteq D_X$, $D_F^* \subseteq D_F$. Конечномерные аппроксимации этих множеств
 обозначаем Θ^X , Θ^F соответственно и называем архивными.

2. Базовые методы решения МКО-задачи

2.1. Метод взвешенных сумм

Классический метод взвешенных целевых функций основан на использовании фитнес-функция вида

$$\varphi(X) = \varphi(\Lambda, F) = \sum_{k=1}^{|F|} \lambda_k f_k(X) = (\Lambda, F),$$

где вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_{|F|})$ представляет собой $|F|$ -мерный вектор весовых коэффициентов $\lambda_k \in [0; 1]$, $k \in [1:|F|]$; $(\cdot, \cdot)$ - символ скалярного произведения. Решение задачи глобальной условной оптимизации

$$\min_{X \in D_X} \varphi(X) = \varphi(X^*) = \varphi(\Lambda, F^*) \quad (2)$$

дает точку X^* на множестве Парето задачи (1) и соответствующую точку F^* на фронте Парето этой задачи.

Схема WS-метода включает в себя следующие основные шаги.

1) Множество $D_\Lambda = \{\Lambda \mid 0 \leq \lambda_k \leq 1, k \in [1:|F|]\}$ допустимых значений вектора весовых множителей Λ покрываем некоторой сеткой Ω_Λ с узлами $\Lambda_j = (\lambda_{1,j}, \lambda_{2,j}, \dots, \lambda_{|F|,j})$, $j \in [1:|\Omega_\Lambda|]$.

2) Для каждого $\Lambda_j \in \Omega_\Lambda$ решаем задачу глобальной условной оптимизации (2) – получает точку $X_j^* \in D_X^*$, а также соответствующую точку $F_j^* \in D_F^*$.

Совокупности $\{X_j^*\} = \Theta^X$, $\{F_j^*\} = \Theta^F$ полученных точек образует искомые аппроксимации множества и фронта Парето соответственно.

Достоинствами WS-метода являются идейная и реализационная простота. Основные недостатки метода состоят в невозможности

аппроксимации невыпуклых частей фронта Парето и отсутствии гарантий получения равномерной аппроксимации.

2.3. Базовый метод доверительной области

Поясним суть *базового метода доверительной области* (*Basic Trust Region, BTR*) [5, 6] на примере задачи безусловной однокритериальной оптимизации

$$\min_{X \in R^{|X|}} f(X) = f(X^*) = f^*. \quad (3)$$

Содержание основных процедур метода заключается в следующем.

Инициализация. Определяем значения свободных параметров метода: δ^0 - начальный радиус области доверия; η_1, η_2 - параметры для оценки адекватности метамоделей функции $f(X)$, $0 < \eta_1 \leq \eta_2 < 1$; γ_1, γ_2 - параметры изменения радиуса области доверия, $0 < \gamma_1 \leq \gamma_2 < 1$. Случайным образом выбираем центральную точку X_C^0 .

Формирование метамоделей. Метамоделей критериальной функции представляет собой аппроксимацию $m^t(X)$ функции $f(X)$ в пределах текущей области доверия (*trust region*)

$$D_C^t = \{X \mid X \in R^{|X|}, \|X - X_C^t\| \leq \delta^t\}.$$

Здесь δ^t - радиус области доверия на текущей итерации t ; $\|\cdot\|$ - некоторая векторная норма, определяющая форму области доверия. Тип «наилучшей» метамоделей зависит от ландшафта критериальной функции. Чаще всего в качестве метамоделей $m^t(X)$ используют квадратичную аппроксимацию функции $f(X)$ в пределах области доверия $D_C^t(X)$. При этом возникает задача построения эффективного плана эксперимента [7], на основе которого строится эта аппроксимация. Обычно используют центральный композиционный план на основе дробной реплики. Авторы считают

перспективным генерацию плана эксперимента на основе $ЛП_\tau$ последовательности [8].

Решение оптимизационной задачи. Данная процедура предполагает решение задачи оптимизации

$$\min_{X \in D_C^t} m^t(X) = m^t(\hat{X}^t), \quad (4)$$

где $\hat{X}^t$ - пробная точка (trial point).

В работе [5] показано, что для сходимости метода *BTR* необязательно решение задачи (4) как задачи глобальной оптимизации – достаточно лишь, чтобы это решение обеспечивало «существенное уменьшение» (*sufficient reduction*) значения функции $m^t(X)$ по сравнению с ее значением в точке X_C^t . На этом основании для сокращения вычислительных затрат задачу (4) сводят к однопараметрической задаче поиска минимума функции $m^t(X)$ вдоль дуги Коши (*Cauchy arc*):

$$\min_{s \geq 0} m^t(X_{Cauchy}^t(s)) = m^t(X_{Cauchy}^t). \quad (5)$$

Здесь $X_{Cauchy}^t(s) = \{X \mid X = X_C^t - s \nabla^T f(X_C^t), s \geq 0, X \in D_C^t\}$ - параметрически заданная дуга Коши, X_{Cauchy}^t - точка Коши.

Схема (5), по сути, является вариацией метода наискорейшего спуска и поэтому может обеспечить лишь линейную скорость сходимости. Более эффективные методы решения задачи (4) основаны на использовании матрицы Гессе $H(X_C^t)$ для выбора направления поиска.

Оценка пробной точки. В качестве количественной оценки ρ^t адекватности построенной модели $m^t(X)$ целевой функции $f(X)$ используем отношение *реального* уменьшения функции f при переходе из точки X_C^t в пробную точку $\hat{X}^t$, *предсказанную* моделью:

$$\rho^t = \frac{f(X_C^t) - f(\hat{X}^t)}{m^t(X_C^t) - m^t(\hat{X}^t)}.$$

Если $\rho^t > \eta_1$, то модель $m^t(X)$ полагаем адекватной и в качестве новой центральной точки принимаем пробную точку, то есть, полагаем $X_C^{t+1} = \hat{X}^t$. В противном случае центральная точка остается прежней: $X_C^{t+1} = X_C^t$.

Выбор радиуса области доверия. Итерацию, для которой $\rho^t \geq \eta_1$, называют *удачной*; $\rho^t \geq \eta_2$ - *очень удачной*; $\rho^t < \eta_1$ - *неудачной*. Напомним, что $\eta_2 \geq \eta_1$. В случае очень удачной итерации радиус доверительной области увеличиваем, то есть, в качестве δ^{t+1} принимает некоторое значение из интервала $[\delta^t; \infty)$. Если итерация t была удачной ($\eta_1 \leq \rho^t \leq \eta_2$), то величину δ^{t+1} уменьшаем или оставляем неизменной: $\delta^{t+1} \in [\gamma_2 \delta^t; \delta^t]$. Наконец, в случае неудачной итерации, радиус уменьшаем: $\delta^{t+1} \in [\gamma_1 \delta^t; \gamma_2 \delta^t]$.

В качестве критерия останова итерационного процесса принимаем один из классических критериев, например, критерий, основанный на евклидовой норме градиента:

$$\|\nabla^T f(X_C^t)\|_2 < \varepsilon.$$

Здесь ε - заданная малая положительная константа.

Известны модификации метода *BTR*, ориентированные на решение задачи условной оптимизации вида (3), когда $X \in D_X \subset R^{|X|}$ [5].

2.3. Канонический метод адаптивных взвешенных сумм

Канонический метод адаптивных взвешенных сумм (*Canonical Adaptive Weighted Sum, CAWS*) предназначен для решения двухкритериальной задачи Парето-аппроксимации и ориентирован на устранение недостатков, присущих *WS*-методу. Метод предложили Ким и Вик (*I.Y. Kim, O.L. de Weck*) [9]. Рассмотрим основные процедуры метода.

Построение грубой Парето-аппроксимации. Грубую аппроксимацию множества Парето строим посредством классического *WS*-метода. Взвешенную сумму критериев записываем в виде

$$\varphi(X) = \lambda f_1(X) + (1 - \lambda) f_2(X), \quad \lambda \in [0; 1].$$

Шаг $\Delta\lambda$ выбираем, исходя из заданного числа разбиений n_0 :

$$\Delta\lambda = \frac{1}{n_0}.$$

Удаление близких решений. В качестве меры близости решений в критериальном пространстве используем евклидову норму: точки Θ_i^F и Θ_j^F текущей аппроксимации фронта Парето полагаем близкими, если

$$\|\Theta_i^F - \Theta_j^F\|_2 < \varepsilon_0,$$

где ε_0 - заданная малая положительная константа. Одну из близких точек исключаем из архивного множества Θ^F и соответствующую точку Θ_i^X или Θ_j^X исключаем из множества Θ^X .

Определение числа разбиений. Число разбиений для i -го сегмента текущей аппроксимации фронта Парето определяет формула

$$n_i = \text{round}\left(C \cdot \frac{l_i}{l_{avg}}\right), \quad l_{avg} = \frac{\sum_{j=1}^m l_j}{m},$$

где n_i и l_i - число разбиений и длина i -ого сегмента соответственно; l_{avg} - средняя длина сегмента; C - константа метода, определяющая плотность разбиения; $\text{round}(\cdot)$ - оператор округления до ближайшего целого; $m = |\Theta^F| - 1$ - число сегментов в текущей аппроксимации фронта Парето.

Решение локальных задач Парето-аппроксимации. Построение аппроксимации фронта Парето в i -ом сегменте состоит в решении МКО-задачи вида (1) в области

$$(D_X)_i = \{X \mid X \in D_X, f_1(X) \leq f_{1,i}^1 - \delta_i^1, f_2(X) \leq f_{2,i}^2 - \delta_i^2\},$$

где δ_i^1, δ_i^2 - *смещения* i -го сегмента текущей аппроксимации фронта Парето [9]. Для решения этой задачи используется WS-метод с шагом

$\Delta\lambda_i = \frac{1}{n_i}$. Полученные точки добавляем в архивы Θ^F , Θ^X .

Итерации метода AWS заканчиваются, когда длины всех сегментов оказываются меньше порогового значения:

$$l_i < \delta^F, \quad i \in [1:|\Theta^F|-1].$$

2.4. Метод адаптивных взвешенных сумм

Метод адаптивных взвешенных сумм (*Adaptive Weighted Sum, AWS*) для двухкритериальной задачи Парето-аппроксимации предложили, как отмечалось выше, Рю, Ким и Ван (*J.-H. Ryu, S. Kim, H. Wan*) в работе [3]. Метод AWS сочетает в себе элементы метода *WS* и метода *BTR*. Основными в методе AWS являются следующие процедуры.

Инициализация метода. Определяем значения свободных параметров метода: δ^0 - начальный радиус области доверия (*trust region radius*); $\rho \in (0; 1)$ - коэффициент сужения этой области; $\delta_{\min}$ - минимальная величина радиуса. Случайным образом в области D_X выбираем центральную точку X_C^0 .

Определение центральной точки. На итерации $(t+1)$ центральную точку $X_C^{t+1} = X'_C$ отыскиваем среди точек текущей Парето-аппроксимации $\Theta^X = \Theta^X(t)$. С этой целью элементы множеств Θ^X , Θ^F сортируем по возрастанию первого частного критерия $f_1(X)$ и представляем в виде линейных списков. Расстояние d_j архивной точки F_j^Θ до ближайших к ней в списке Θ^F точек определяет формула

$$d_j = \|F_{j-1}^\Theta - F_j^\Theta\|_2 + \|F_j^\Theta - F_{j+1}^\Theta\|_2, \quad j \in [1:|\Theta|]. \quad (6)$$

Алгоритм определения центральной точки X'_C использует следующее правило.

1) Если $|\Theta| > 2$, то полагаем $X'_C = X_{j^*}^\Theta$, где

$$j^* = \arg \left(\max_{j \in [2:|\Theta|-1]} d_j \mid X_j^\Theta \notin X^t \right).$$

Здесь X^t - множество точек, использованных в качестве центральных на всех предыдущих итерациях $[0: t]$. Иными словами, за центральную точку принимаем точку, во-первых, наиболее удаленную от других точек множества Θ^X в смысле расстояния (6), и, во-вторых, не использованную на предшествующих итерациях.

2) Если $|\Theta| = 2$, то с равной вероятностью полагаем $X'_C = X_1^\Theta$ или $X'_C = X_2^\Theta$.

3) Если $|\Theta| = 1$, то принимается $X'_C = X_1^\Theta$.

Формирование метамоделей. Мета модель $m_i^{t+1}(X) = m'_i(X)$ представляет собой квадратичную аппроксимацию функции $f_i(X)$ в окрестности точки X'_C :

$$\begin{aligned} m'_1(X) &= f_1(X'_C) + \nabla^T f_1(X'_C)(X - X'_C) + \frac{1}{2}(X - X'_C)^T H_1(X'_C)(X - X'_C); \\ m'_2(X) &= f_2(X'_C) + \nabla^T f_2(X'_C)(X - X'_C) + \frac{1}{2}(X - X'_C)^T H_2(X'_C)(X - X'_C). \end{aligned}$$

Здесь $\nabla f_i(X'_C)$, $H_i(X'_C)$ - вектор градиента и матрица Гессе функции $f_i(X)$ в точке X'_C .

Если $|\Theta| > 2$, то дополнительно строим мета модели

$$m'_p(X) = \lambda_1^p m'_1(X) + \lambda_2^p m'_2(X),$$

$$m'_q(X) = \lambda_1^q m'_1(X) + \lambda_2^q m'_2(X),$$

а если $|\Theta| = 2$ или $|\Theta| = 1$ - мета модель

$$m'_p(X) = \lambda_1^p m'_1(X) + \lambda_2^p m'_2(X).$$

В случае $|\Theta| > 2$ весовые множители $(\lambda_1^p, \lambda_2^p) = \Lambda^p$, $(\lambda_1^q, \lambda_2^q) = \Lambda^q$ определяем по правилу

$$\Lambda^p = c^p \left((-f_2(X'_C) + f_2(X_{j^*-1}^\Theta)), (f_1(X'_C) - f_1(X_{j^*-1}^\Theta)) \right),$$

$$\Lambda^q = c^q \left((-f_2(X_{j^*+1}^\Theta) + f_2(X'_C)), (f_1(X_{j^*+1}^\Theta) - f_1(X'_C)) \right);$$

в случае $|\Theta| = 2$ – по правилу

$$\Lambda^p = c^p \left((-f_2(X_2^\Theta) + f_2(X_1^\Theta)), (f_1(X_2^\Theta) - f_1(X_1^\Theta)) \right);$$

в случае $|\Theta| = 1$ – по правилу $\Lambda^p = (0,5, 0,5)$. Здесь константы c^p, c^q выбираются таким образом, чтобы обеспечить выполнение условий нормировки $\lambda_1^p + \lambda_2^p = \lambda_1^q + \lambda_2^q = 1$.

В работе [3] процедура формирования квадратичных метамоделей $m'_1(X), m'_2(X)$ не раскрыта. Остановимся на этом вопросе.

Для формирования квадратичных метамоделей $m'_1(X), m'_2(X)$ используем *центральный композиционный план* (ЦКП) с центром в точке X'_C [7]. Для упрощения вычислений «звездное» плечо в ЦКП выбираем таким образом, чтобы этот план приобрел свойство ортогональности. В качестве ядра ЦКП используем полный факторный эксперимент. Легко видеть, что в общем случае необходимое число испытаний (вычислений значений критериальной функции) для построения одной метамоделей равно

$$n = n_k + 2|X| + n_0,$$

где n_k – число испытаний в точках ядра плана, $2|X|$ – число «звездных» точек, $n_0 = 1$ – число испытаний в центре плана. Таким образом, в нашем случае (когда $|X| = 2$) имеем

$$n = 2^2 + 2 \cdot 2 + 1 = 9.$$

Серьезной проблемой при использовании ЦКП является генерация эффективной дробной реплики. Используем для этой цели функции Уолша [10].

Решение оптимизационных задач. Данная процедура предполагает решение задач оптимизации

$$\min_{X \in D'_C} m'_1(X) = m'_1(\hat{X}'_1), \quad \min_{X \in D'_C} m'_2(X) = m'_2(\hat{X}'_2), \quad (7)$$

где текущую область доверия D'_C определяет формула

$$D'_C = \{X \mid X \in D_X, \|X - X'_C\| \leq \delta^t\}.$$

Если $|\Theta| > 2$, то решения $\hat{X}'_1, \hat{X}'_2$ позволяют отыскать приближенно оптимальные по Парето точки $\hat{X}'_p, \hat{X}'_q$, принадлежащие области доверия D'_C , путем решения оптимизационных задач

$$\min_{X \in D'_C} m'_p(X) = m'_p(\hat{X}'_p), \quad \min_{X \in D'_C} m'_q(X) = m'_q(\hat{X}'_q). \quad (8)$$

Данный этап алгоритма иллюстрирует рисунок 1, на котором принято

$$F'_C = F(X'_C), \quad F_{j^*_{-1}}^{\ominus} = F(X_{j^*_{-1}}^{\ominus}), \quad F_{j^*_{+1}}^{\ominus} = F(X_{j^*_{+1}}^{\ominus}).$$

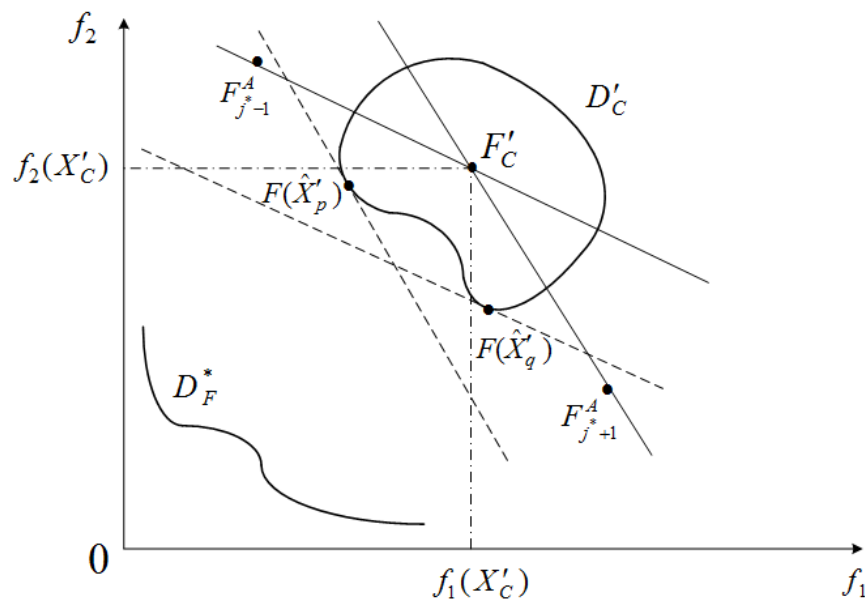


Рисунок 1 – К схеме метода AWS: результаты решения задач (7)

Важно, что задачи (7), (8) представляют собой задачи оптимизации квадратичных функций, для решения которых известны высокоэффективные методы, алгоритмы и соответствующее программное обеспечение.

В процессе итераций текущий радиус области доверия уменьшаем по правилу $\delta^{t+1} = \rho \delta^t$ до достижения минимально допустимой его величины $\delta_{\min}$. Новое состояние архивного множества Θ^X получаем путем добавления в него точек $\hat{X}'_1$, $\hat{X}'_2$, $\hat{X}'_p$, $\hat{X}'_q$ и исключения из полученного набора

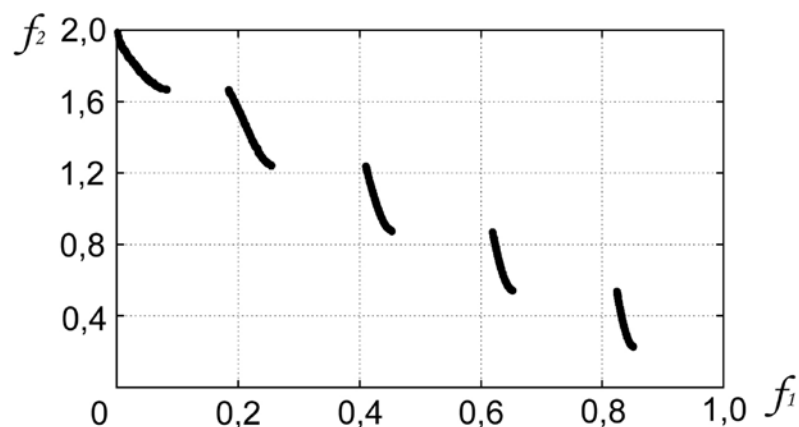
доминируемых решений. Аналогично, множество Θ^F формируем путем добавления в него точек $F(\hat{X}'_1)$, $F(\hat{X}'_2)$, $F(\hat{X}'_p)$, $F(\hat{X}'_q)$.

3. Модификации метода адаптивных взвешенных сумм

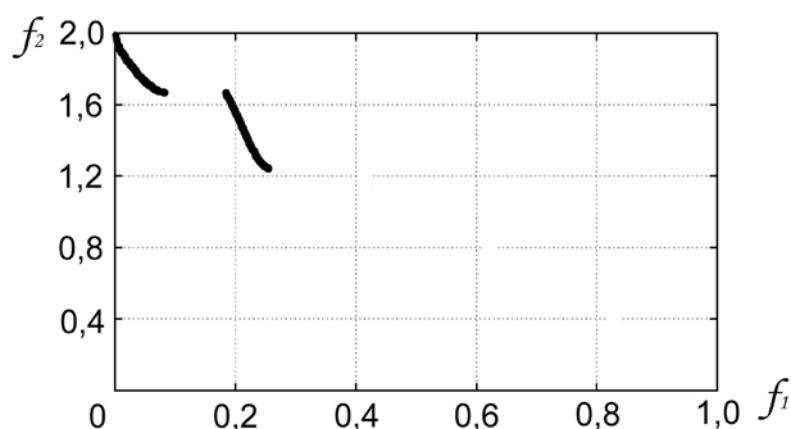
3.1. Повышение разнообразия множества архивных точек

(модификация AWS^{M1})

Наши исследования показали, что в некоторых случаях метод AWS обеспечивает аппроксимацию только части множества Парето, то есть не обеспечивает достаточное разнообразие множества архивных точек Θ^x (рисунок 2).



а) точный фронт Парето



б) результат аппроксимации фронта методом AWS

Рисунок 2 – Тестовая задача ZDT3 (п. 4.1)

Суть модификации AWS^{M1} состоит в решении на каждой итерации в крайних точках архивного множества Θ^X задач

$$\begin{aligned} \min_{X \in D'_1} m'_1(X) &= m'_1(\hat{X}'_1), & \min_{X \in D'_1} m'_2(X) &= m'_2(\hat{X}'_2); \\ \min_{X \in D'_2} m'_1(X) &= m'_1(\hat{X}'_1), & \min_{X \in D'_2} m'_2(X) &= m'_2(\hat{X}'_2). \end{aligned}$$

Здесь

$$D'_1 = \{X \mid X \in D_X, \|X - X'_1\| \leq \delta_1\}, \quad D'_2 = \{X \mid X \in D_X, \|X - X'_2\| \leq \delta_2\}.$$

3.2. Смещение области доверия (модификация AWS^{M2})

Точки множества Парето могут в некоторых случаях плотно прилегать к одной из границ области определения (рисунок 3). При решении таких задач метод AWS уменьшает радиус области доверия D'_C на каждой итераций, сокращая в результате разнообразие архивных точек.

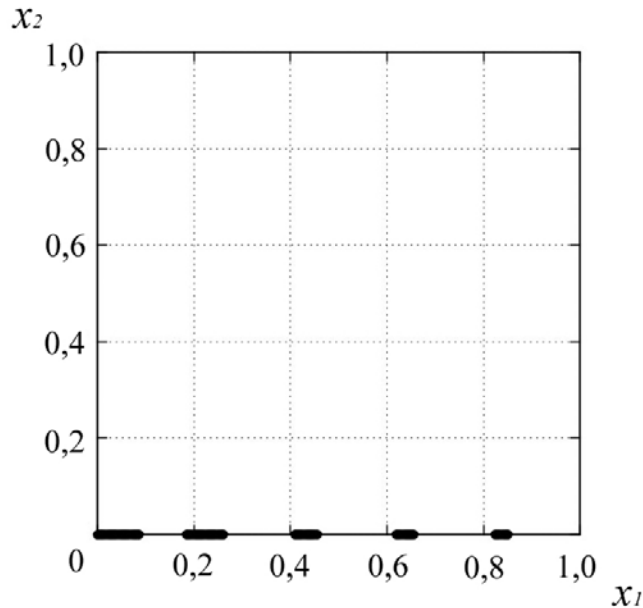


Рисунок 3 – Множество Парето тестовой задачи $ZDT3$ (п. 5.1): $|X| = 2$

Идея модификации AWS^{M2} состоит в смещении центра области доверия «вглубь» области определения, не изменяя при этом ее радиуса (рисунок 4).

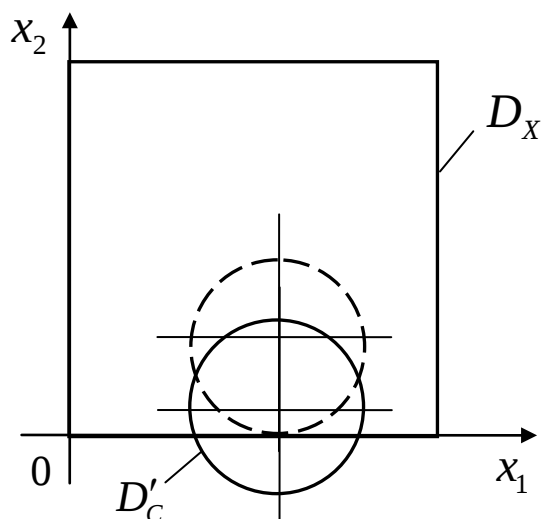


Рисунок 4 – Смещение области доверия: $|X| = 2$

3.3. Нейросетевая аппроксимация целевых функций (модификация AWS^{M3})

Идея данной модификации состоит в построении метамоделей критериальных функций с помощью нейронных сетей [11]. Поскольку на каждой итерации метода AWS аппроксимация критериальных функций производится в пределах «небольшой» области доверия D'_C используем радиально-базисные нейронные сети. Полагаем число нейронов N в скрытом слое равным девяти (по числу точек испытаний в ЦКП) и располагаем их в указанных точках. Для обучения нейронной сети используется алгоритм Нелдера-Мида, варьируемыми параметрами которого являются веса w_i и ширины σ_i радиально-базисных функций; $i \in [1: N]$.

4. Программная реализация

Для реализации метода AWS и его модификаций AWS^{M1} - AWS^{M4} использовано программное обеспечение, которое является свободным и распространяется по лицензии *GPL (GNU Public License)*. В силу кросс-платформенности и высокой эффективности программная реализация выполнена на языке программирования $C++$. Для компиляции использовался

набор компиляторов *GCC (GNU Compiler Collection)*. Разработанное программное обеспечение использует библиотеку численного анализа *GNU Scientific Library (GSL)* и аналогичную библиотеку *F2C (Fortran to C)* для трансляции программного кода, написанного на языке Fortran, в код на языке C. Графический вывод результатов работы программы реализован с помощью утилиты *Gnuplot*.

Основными являются следующие классы программной модели:

- *AWSAlgorithm* – реализация алгоритма AWS;
- *OptimAlgorithm*, *PenaltyAlgorithm*, *SlaceTolAlgorithm* – алгоритмы условной оптимизации при решении однокритериальных задач (7), (8);
- *ApproximatorNNet*, *ApproximatorCCD* – аппроксимация критериальных функций;
- *Design*, *FullFactorialDesign*, *WalshDesign* – планирование экспериментов для квадратичной аппроксимации целевых функций;
- *HyperSphereConstrain*, *LinearConstrain* – формирование ограничений;
- *ParetoBruteForcer* – генерация начальной аппроксимации множества Парето;
- *ParetoMeasurer* – оценка качества аппроксимации множества (фронта).

Программное обеспечение реализует планы полного факторного эксперимента и дробной реплики на основе функций Уолша (классы *FullFactorialDesign* и *WalshDesign* соответственно). Интерфейс абстрактного класса *Design* позволяет использовать любой метод локальной условной оптимизации. Метод наименьших квадратов реализован с использованием библиотеки *GSL* (функция *gsl_multifit()*). Решение задач локальной условной однокритериальной оптимизации (7), (8) осуществляется методами штрафных функций и скользящего допуска в комбинации с известным методом Нелдера-Мида (классы *PenaltyAlgorithm* и *SlaceTolAlgorithm* соответственно). Интерфейс абстрактного класса *OptimAlgorithm* позволяет использовать любой метод локальной условной оптимизации.

Общий объем программного кода составляет около 5000 строк.

5. Исследование эффективности модификаций метода адаптивных взвешенных сумм

5.1. Организация экспериментов

Индикаторы качества алгоритма. Качество аппроксимации множества Парето оцениваем с помощью следующих индикаторов [12]:

- мощность множества решений (*overall non-dominated vector generation*)

$$I_{ONVG}(\Theta) = |\Theta| \rightarrow \max;$$

- близость найденных решений к точному множеству Парето рассматриваемой МКО-задачи (*generational distance*)

$$I_{GD} = \frac{\sqrt{\sum_{i=1}^{|\Theta|} \varepsilon_i^2}}{I_{ONVG}(\Theta)} \rightarrow \min,$$

где ε_i - расстояние от i -ой точки архива недоминируемых решений Θ до ближайшего точного решения;

- равномерность распределения решений в полученной Парето-аппроксимации (*spacing*)

$$I_{SP} = \sqrt{\frac{1}{I_{ONVG}-2} \sum_{i=1}^{|\Theta|-1} (\bar{s} - s_i)^2} \rightarrow \min,$$

где s_i - расстояние от i -ой точки архива Θ до ближайшей соседней точки; $\bar{s}$ - среднее значение s_i .

В качестве индикатора вычислительной эффективности метода используем число испытаний n_E .

Число итераций в вычислительных экспериментах ограничиваем величиной $\hat{t}$.

Тестовые МКО задачи. Вычислительные эксперименты выполнены с использованием следующих тестовых задач.

- 1) Двухмерная двухкритериальная задача Аудета (C. Audet):

$$D_X = \{X \mid 0 \leq x_i \leq 1, i = 1, 2\};$$

$$f_1(X) = f_1(x_1) = 4 \cdot x_1,$$

$$f_2(X) = g(X) \cdot h(X);$$

$$h(X) = \begin{cases} 1 - \left(\frac{f_1(X)}{g(X)} \right)^\alpha, & f_1(X) \leq g(X), \\ 0, & \text{иначе,} \end{cases}$$

$$g(X) = g(x_2) = 4 - 3 \cdot \exp \left(- \left\{ \frac{x_2 - 0,2}{0,02} \right\}^2 \right).$$

Фронт Парето задачи является непрерывным; значению $\alpha = 0,25$ соответствует выпуклый фронт, а значению $\alpha = 4$ - невыпуклый [13].

2) Двумерная двухкритериальная задача ZDT3 [12]

$$D_X = \{X \mid 0 \leq x_i \leq 1, i \in [1:|X|], |X| = 30\},$$

$$f_1(X) = f(x_1) = x_1,$$

$$f_2(X) = g(X) \cdot \left[1 - \sqrt{\frac{f_1(X)}{g(X)}} - \frac{f_1(X)}{g(X)} \cdot \sin(10 \pi f_1(X)) \right],$$

где

$$g(X) = g(x_2, \dots, x_{|X|}) = 1 + 9 \cdot \sum_{i=2}^{|X|} \frac{x_i}{|X|-1}.$$

Сложность задачи обусловлена несвязным, хотя и выпуклым, фронтом Парето.

3) Двумерная двухкритериальная задача ZDT6 [12]:

$$D_X = \{X \mid 0 \leq x_i \leq 1, i \in [1:|X|], |X| = 10\};$$

$$f_1(X) = f(x_1) = 1 - \exp(-4 x_1) \cdot \sin^6(6 \pi x_1),$$

$$f_2(X) = g(X) \cdot \left[1 - \left(\frac{f_1(X)}{g(X)} \right)^2 \right];$$

$$g(X) = g(x_2, \dots, x_{|X|}) = 1 + 9 \cdot \left(\sum_{i=2}^{|X|} \frac{x_i}{|X|-1} \right)^{0,25}.$$

Сложность заключается в том, что задача имеет невыпуклый фронт Парето.

4) Двумерная двухкритериальная задача ZDT7 [12]:

$$D_X = \{X \mid 0 \leq x_i \leq 1, i \in [1:|X|], |X| = 30\};$$

$$f_1(X) = f(x_1) = x_1,$$

$$f_2(X) = g(X) \cdot h(X);$$

$$g(X) = g(x_2, \dots, x_{|X|}) = 1 + 9 \cdot \sum_{i=2}^{|X|} \frac{x_i}{|X|-1},$$

$$h(X) = 1 - \sqrt{\frac{f_1(X)}{g(x_2, \dots, x_{|X|})}}.$$

Задача имеет непрерывный выпуклый фронт Парето и несколько локальных фронтов.

5.2. Результаты экспериментов

Модификация AWS^{M1} (п. 3.1). Рассматриваем случай $|X| = 2$. Исследование выполнено на двухкритериальных тестовых задачах Аудета и ZDT3. Результаты экспериментов представлены в таблицах 1 - 4 и на рисунках 5 - 14. На этих рисунках и далее черные точки показывают конечное состояние множеств Θ^X , Θ^F , а светлые точки – их промежуточные состояния. Для задачи ZDT3 представляем результаты двух экспериментов (старты 1, 2).

Таблица 1 – Тестовая задача Аудета (выпуклый фронт): результаты экспериментов

Метод	Индикаторы эффективности			
	n_E	$I_{ONVG}(\Theta)$	I_{GD}	I_{SP}
AWS^{M1}	637	70	0,202	0,065
AWS	450	33	0,087	0,122

Таблица 2 – Тестовая задача Аудета (невыпуклый фронт): результаты экспериментов

Метод	Индикаторы эффективности			
	n_E	$I_{ONVG}(\Theta)$	I_{GD}	I_{SP}
AWS^{M1}	868	19	0,008	0,052
AWS	450	16	0,028	0,13

Таблица 3 – Тестовая задача *ZDT3*: результаты экспериментов; старт 1

Метод	Индикаторы эффективности			
	n_E	$I_{ONVG}(\Theta)$	I_{GD}	I_{SP}
AWS^{M1}	494	90	0,011	0,04
AWS	450	84	0,012	0,04

Таблица 4 – Тестовая задача *ZDT3*: результаты экспериментов; старт 2

Метод	Индикаторы эффективности			
	n_E	$I_{ONVG}(\Theta)$	I_{GD}	I_{SP}
AWS^{M1}	527	76	0,01	0,045

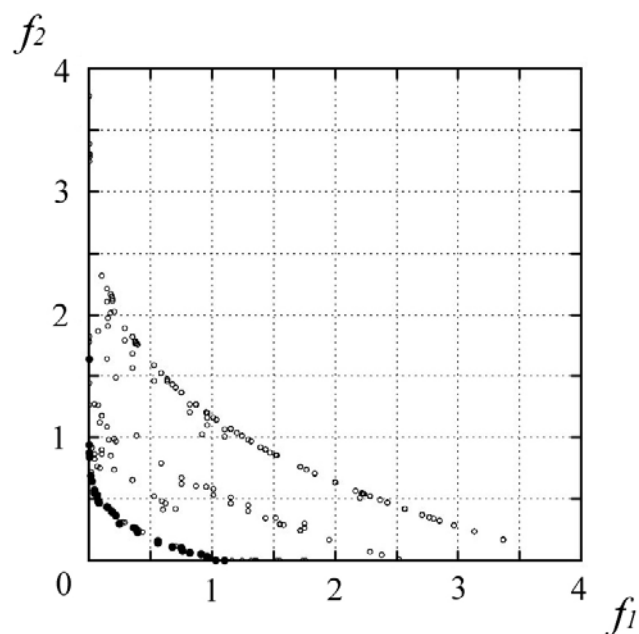


Рисунок 5 – Задача Аудета (выпуклый фронт):

метод AWS^{M1} ; $\delta = 0,2$; $\rho = 2$; $\delta_{\min} = 0,001$; $\hat{t} = 30$

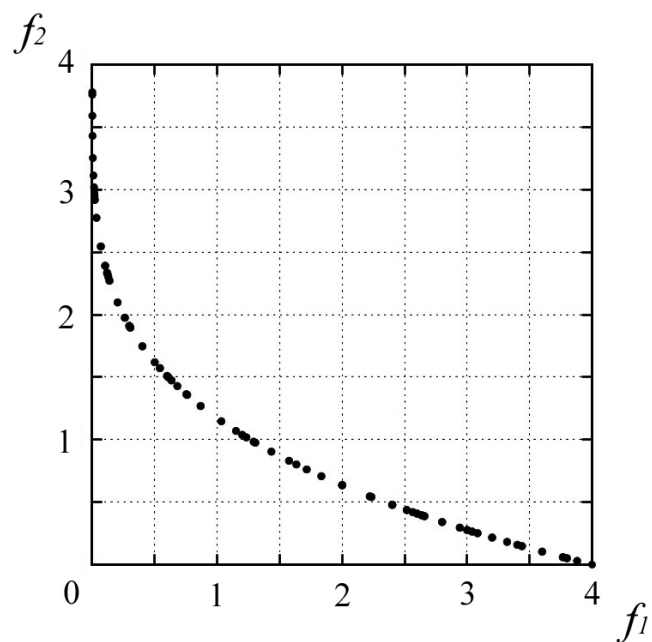


Рисунок 6 – Задача Аудета (выпуклый фронт):
метод AWS; $\delta = 0,2$; $\rho = 2$; $\delta_{\min} = 0,001$; $\hat{t} = 30$

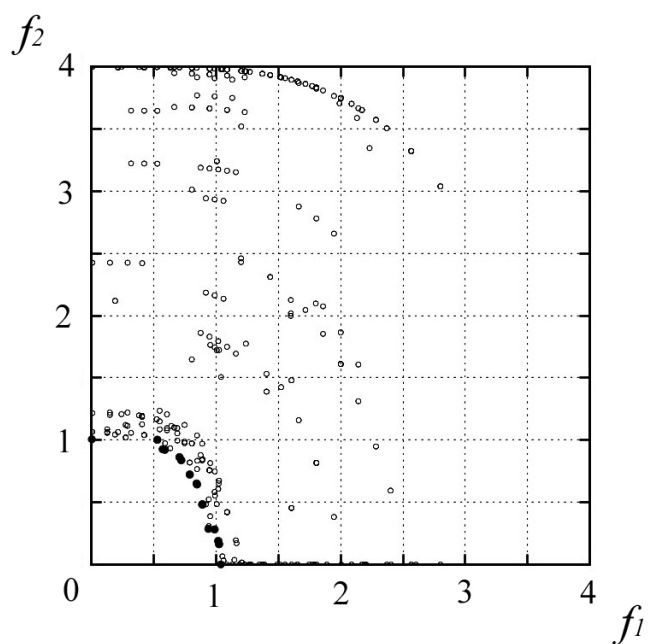


Рисунок 7 – Задача Аудета (невыпуклый фронт);
метод AWS; $\delta = 0,2$; $\rho = 2$; $\delta_{\min} = 0,001$; $\hat{t} = 30$

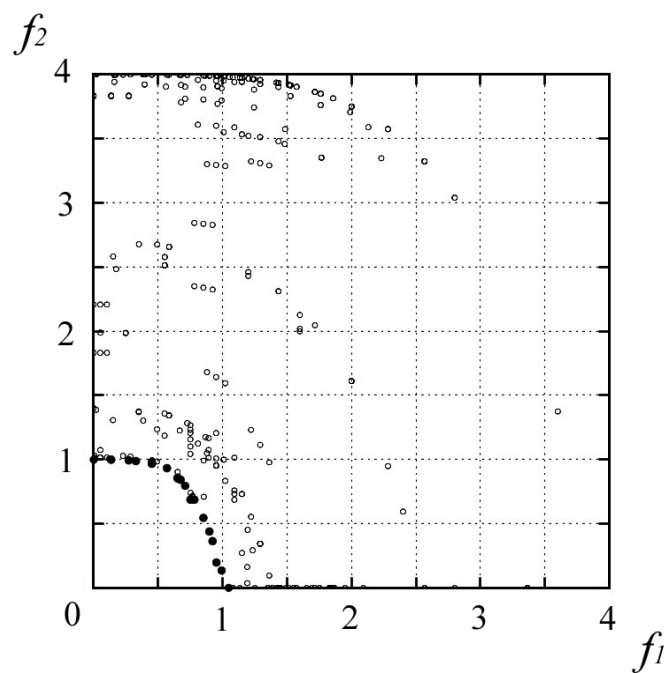


Рисунок 8 – Задача Аудета (невыпуклый фронт):
метод AWS^{M1} ; $\delta = 0,2$; $\rho = 2$; $\delta_{\min} = 0,001$; $\hat{t} = 30$

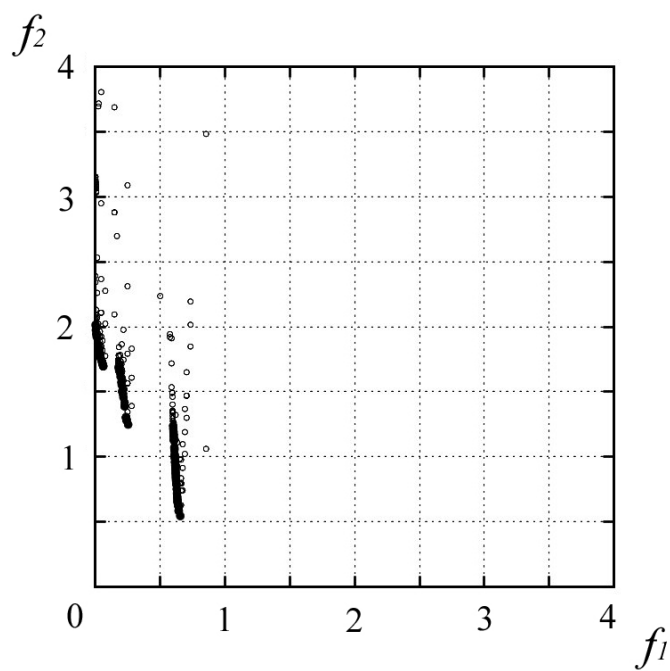


Рисунок 9 – Задача ZDT3: метод AWS ; $\delta = 0,5$; $\rho = 1,5$; $\delta_{\min} = 0,001$; $\hat{t} = 30$;
старт 1

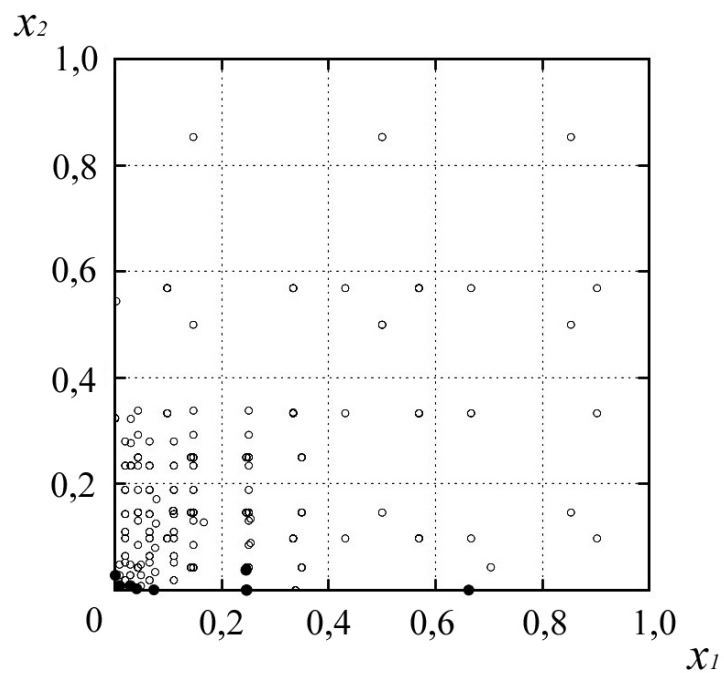


Рисунок 10 – Задача ZDT3 (пространство параметров): метод AWS; $\delta = 0,5$;
 $\rho = 1,5$; $\delta_{\min} = 0,001$; $\hat{t} = 30$; старт 1

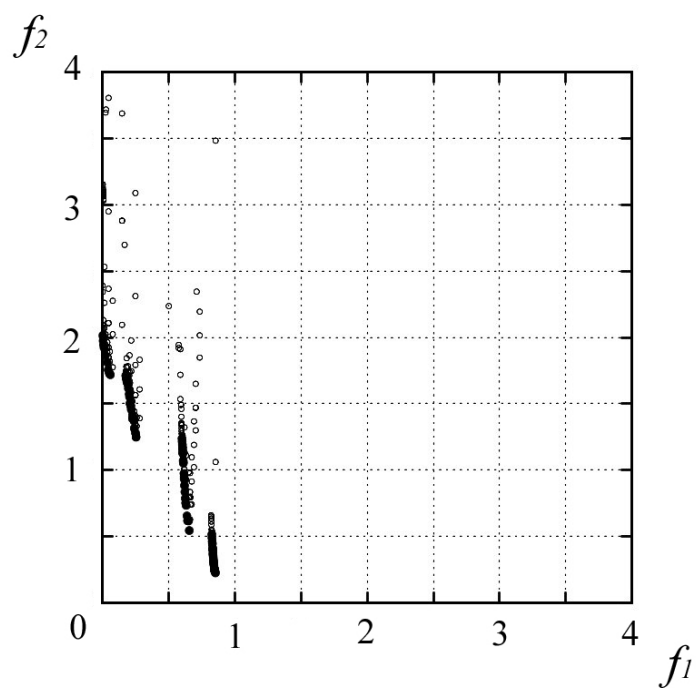


Рисунок 11 – Задача ZDT3: метод AWS^{M1} ; $\delta = 0,5$; $\rho = 1,5$; $\delta_{\min} = 0,001$;
 $\hat{t} = 30$; старт 1

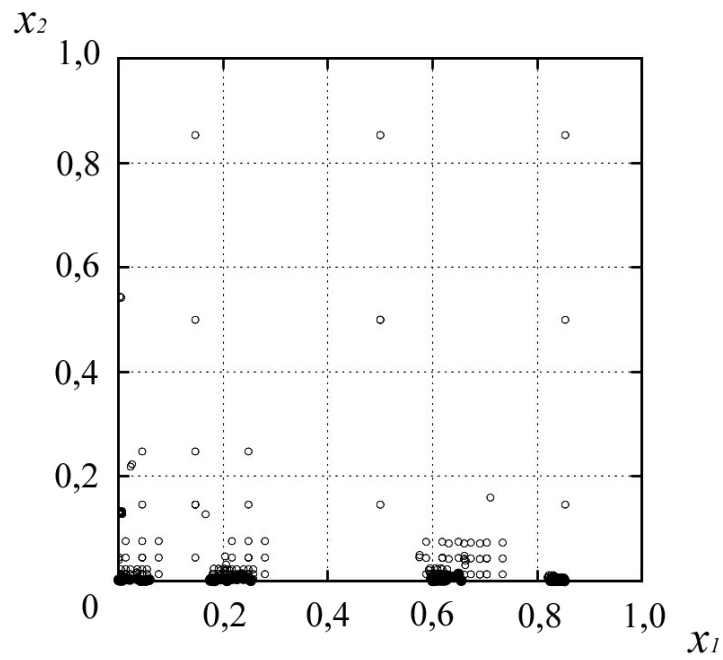


Рисунок 12 – Задача *ZDT3* (пространство параметров): метод AWS^{M1} ;
 $\delta = 0,5$; $\rho = 1,5$; $\delta_{\min} = 0,001$; $\hat{t} = 30$; старт 1

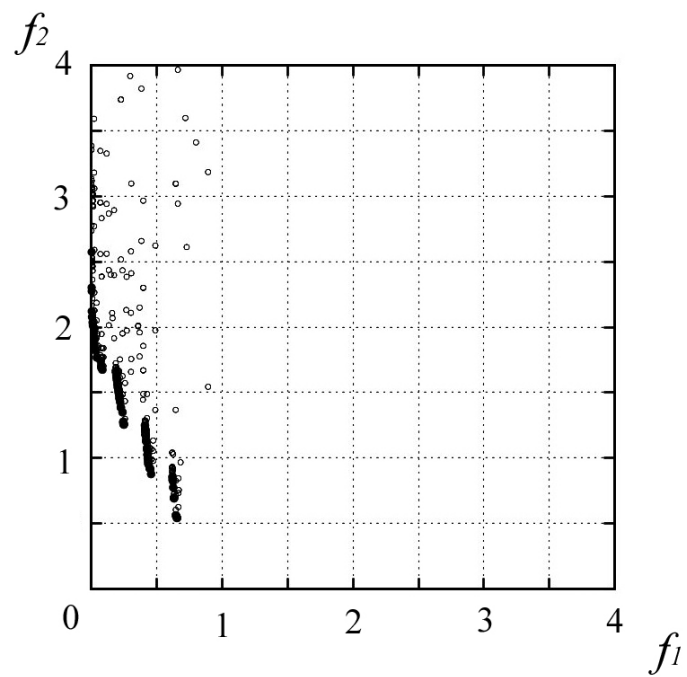


Рисунок 13 – Задача *ZDT3*: метод AWS^{M1} ; $\delta = 0,5$; $\rho = 1,5$; $\delta_{\min} = 0,001$;
 $\hat{t} = 30$; старт 2

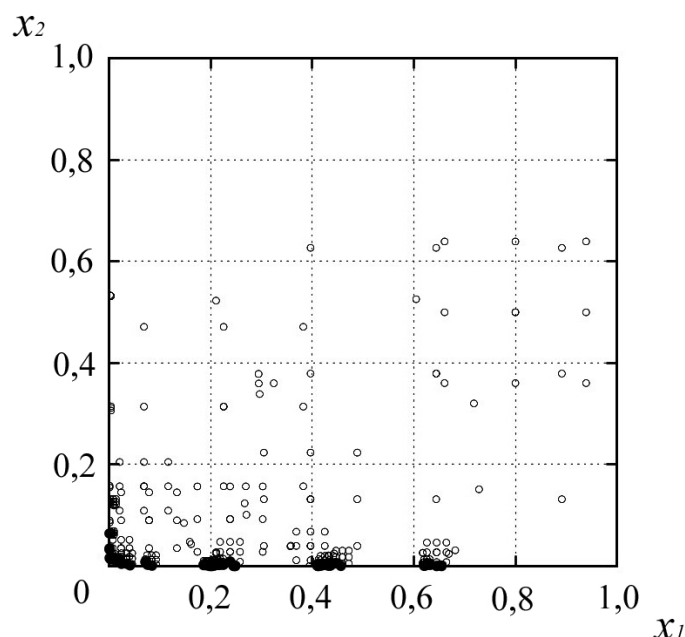


Рисунок 14 – Задача ZDT3 (пространство параметров): метод AWS^{M1} ;

$$\delta = 0,5; \rho = 1,5; \delta_{\min} = 0,001; \hat{t} = 30; \text{старт } 2$$

Представленные результаты показывают, что для задачи Аудета, имеющей выпуклый фронт Парето, модификация AWS^{M1} превосходит метод AWS по индикатору $I_{ONVG}(\Theta)$ более чем в два раза, а по индикатору I_{SP} - почти в два раза. В то же время, модификация AWS^{M1} проигрывает методу AWS по индикатору n_E примерно на 30% и по индикатору I_{GD} - примерно на 60%. Для более сложной задачи Аудета, имеющей невыпуклый фронт Парето, модификация AWS^{M1} обеспечивает близкие к методу AWS результаты по индикатору $I_{ONVG}(\Theta)$ и примерно в 3,5 и 2,5 раз лучшие результаты по индикаторам I_{GD} и I_{SP} соответственно. С другой стороны, для этой задачи метод AWS^{M1} проигрывает методу AWS по индикатору n_E примерно на 50%.

Иная картина наблюдается в случае задачи ZDT3, имеющей, напомним, выпуклый, но разрывный фронт Парето. В этом случае модификация AWS^{M1} по всем рассматриваемым индикаторам обеспечивает результаты, близкие к методу AWS . В то же время, из сравнения результатов, представленных на

рисунках 9, 11, 13, вытекает, что в условиях экспериментов метод AWS локализует три из пяти фрагментов разрывного фронта Парето, а метод AWS^{M1} - четыре. Таким образом, в случае задачи $ZDT3$ метод AWS не обеспечивает удовлетворительное качество Парето-аппроксимации, отыскивая только часть множества и фронта Парето. Модифицированный метод AWS^{M1} справляется с этой задачей лучше, находя большую часть этих множеств.

Модификация AWS^{M2} (п. 3.2). Эффективность данной модификации оценена с помощью двухкритериальной задачи $ZDT3$. Результаты экспериментов при числе итераций $\hat{t} = 30$ представлены на рисунках 15, 16. Эти результаты показывают, что ожидаемое увеличение скорости сходимости в сравнении с методом AWS не достигнуто. Необходимы дополнительные исследования для объяснения этого эффекта.

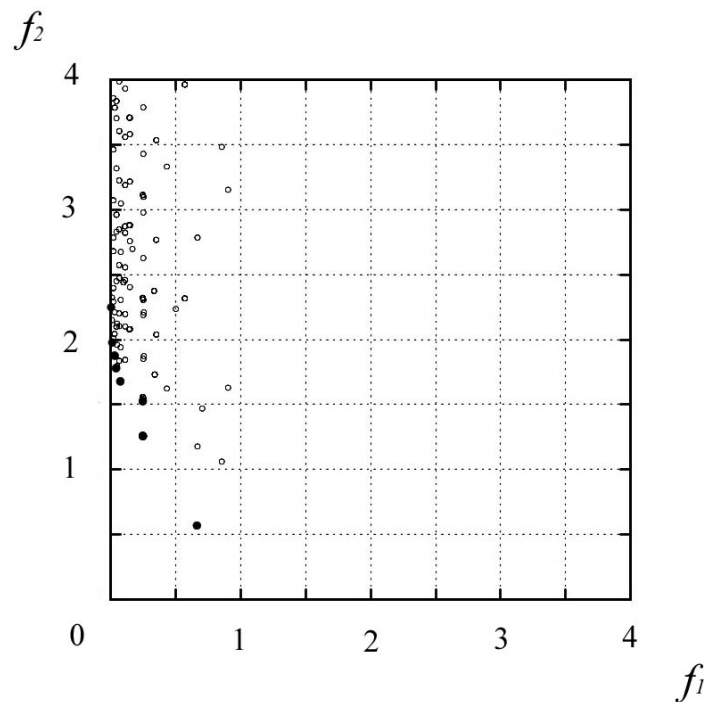


Рисунок 15 – Задача $ZDT3$: метод AWS^{M2} ; $\delta = 0,5$; $\rho = 1,5$; $\delta_{\min} = 0,001$;

$$\hat{t} = 30$$

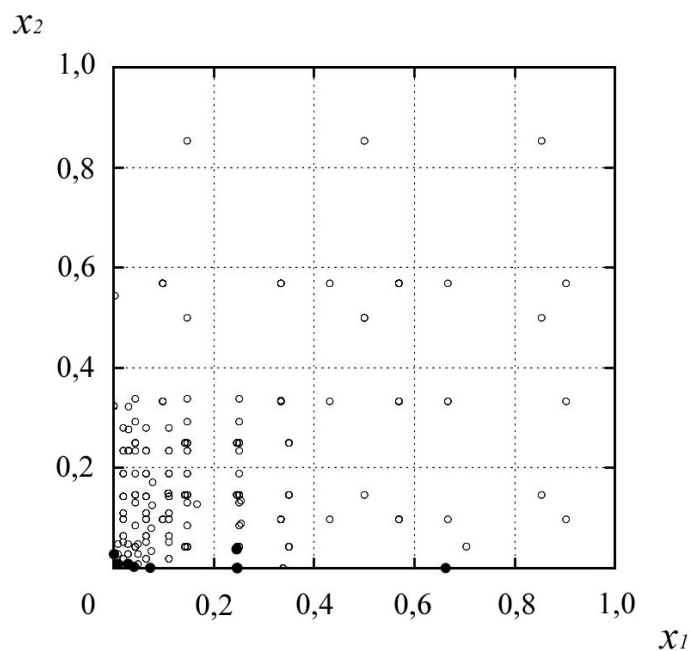


Рисунок 16 – Задача *ZDT3* (пространство параметров): метод AWS^{M2} ;

$$\delta = 0,5; \rho = 1,5; \delta_{\min} = 0,001; \hat{t} = 30$$

Модификация AWS^{M3} (п. 3.3). Исследование выполнено на двухкритериальных тестовых задачах *ZDT3* и *ZDT7* при $|X| = 2$. Результаты экспериментов представлены в таблицах 5, 6 и на рисунках 17 - 20.

Таблица 5 – Тестовая задача *ZDT3*: результаты экспериментов

Метод	Индикатор эффективности			
	n_E	$I_{ONVG}(\Theta)$	I_{GD}	I_{SP}
<i>AWS</i>	10314	438	0,0007	0,017
AWS^{M3}	35536	91	0,005	0,046

Таблица 6 – Тестовая задача *ZDT7*: результаты экспериментов

Метод	Индикаторы эффективности			
	n_E	$I_{ONVG}(\Theta)$	I_{GD}	I_{SP}
<i>AWS</i>	9654	133	0,001	0,015
AWS^{M3}	36384	388	0,0005	0,009

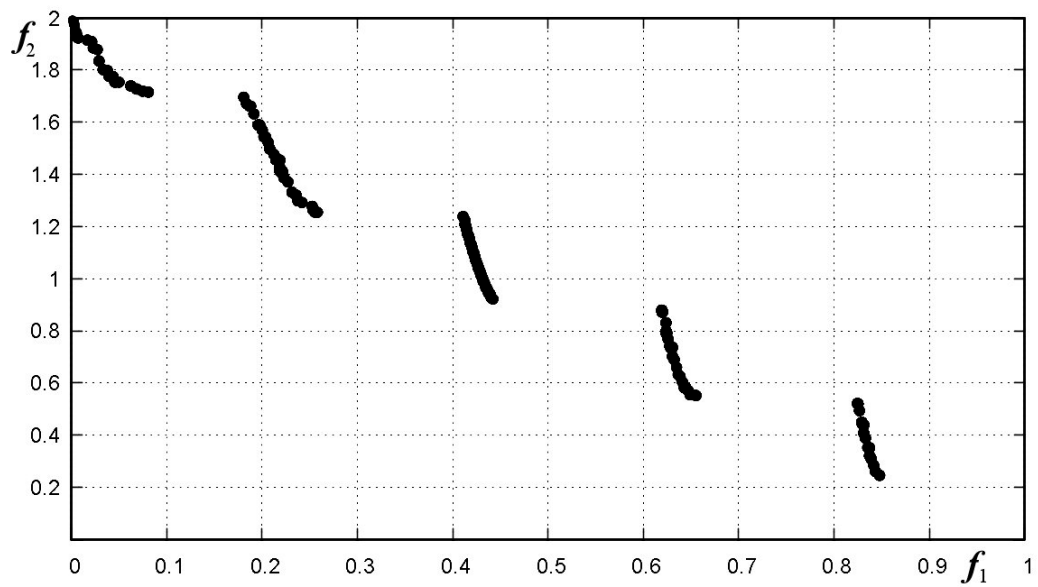


Рисунок 17 – Тестовая задача *ZDT3*: метод *AWS*;

$$\delta = 0,8; \rho = 1,5; \delta_{\min} = 0,001; \hat{t} = 200$$

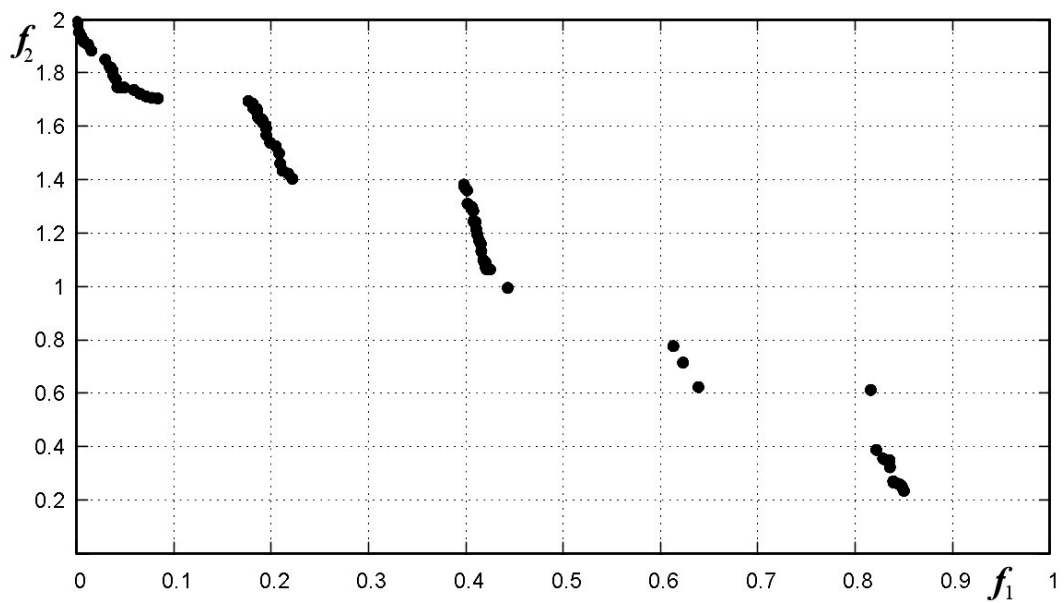


Рисунок 18 – Тестовая задача *ZDT3*: метод *AWS^{M3}*;

$$\delta = 0,8; \rho = 1,5; \delta_{\min} = 0,001; \hat{t} = 200$$

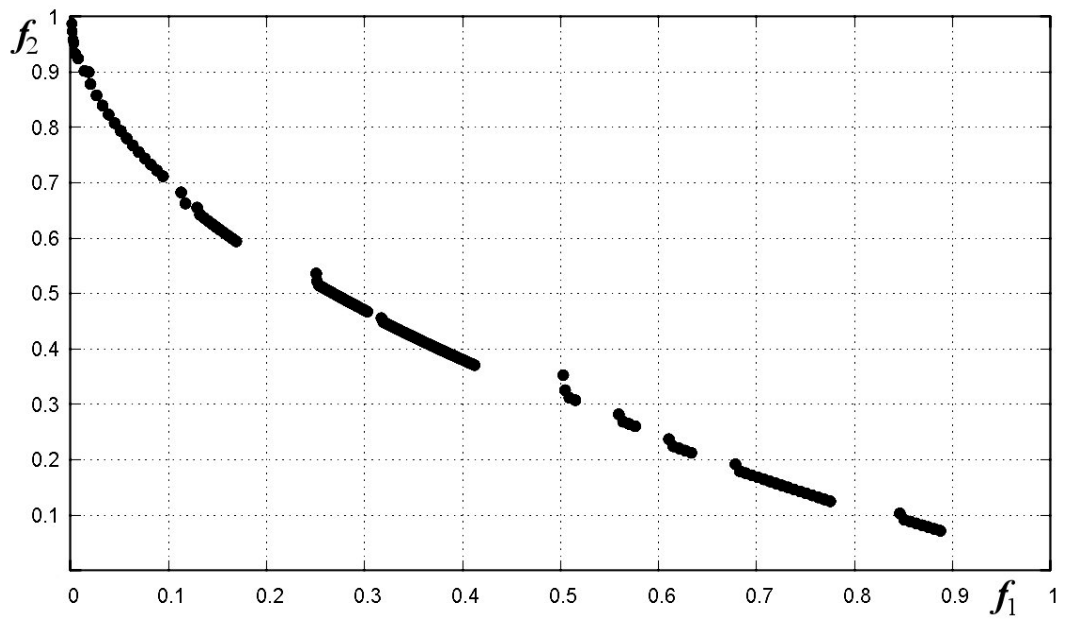


Рисунок 19 – Тестовая задача *ZDT7*: метод *AWS*; $\delta = 0,8$; $\rho = 1,5$;

$$\delta_{\min} = 0,001; \hat{t} = 500$$

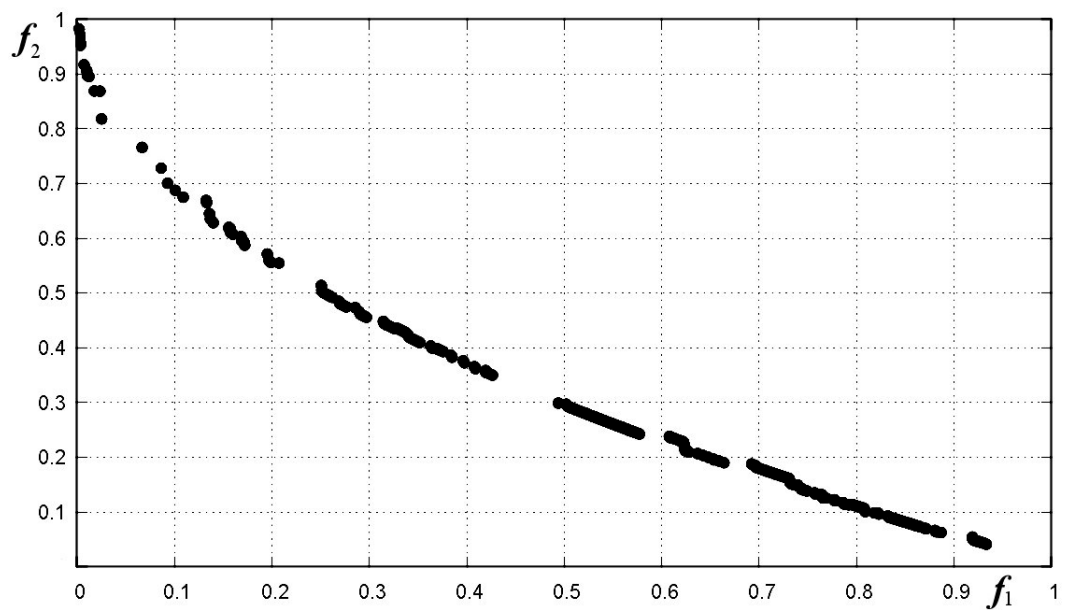


Рисунок 20 – Тестовая задача *ZDT7*: метод *AWS^{M3}*; $\delta = 0,8$; $\rho = 1,5$;

$$\delta_{\min} = 0,001; \hat{t} = 500$$

Представленные результаты показывают, что для задачи *ZDT3* модификация *AWS^{M3}* проигрывает методу *AWS* по всем четырем рассматриваемым индикаторам эффективности. Для задачи *ZDT7* имеет

место иная ситуация: по индикатору $I_{ONVG}(\Theta)$ модификация AWS^{M3} лучше метода AWS почти в три раза, по индикаторам I_{GD} , I_{SP} - в два и 1,7 раза соответственно. С другой стороны, такие высокие результаты достигаются модификацией AWS^{M3} за счет почти четырехкратного увеличения числа испытаний, то есть индикатора n_E .

Заключение

В работе представлен обзор метода AWS , а также методов, на которых он основан. Предложены три модификации метода AWS , имеющие целью устранение его недостатков. Выполнена программная реализация метода AWS и его указанных модификаций. На известных тестовых МКО-задачах выполнено широкое исследование эффективности разработанного методического, алгоритмического и программного обеспечения. Рассмотрены четыре индикатора эффективности, характеризующие, как качество получаемой Парето-аппроксимации, так и соответствующие вычислительные затраты. Исследование показало, что часто предложенные модификации обеспечивают значительно превосходство над методом AWS . В тоже время, исследование не позволило получить все ожидаемые результаты. Авторы предполагают дополнительные исследования для локализации причин неудачных модификаций. Кроме того, авторы планируют исследование эффективности модификаций на тестовых задачах высокой размерности.

Работа поддержана грантом РФФИ № 12-07-00324-а «Структурная и параметрическая идентификация кинетических моделей реакций нейтрального металлокомплексного катализа».

Список литературы

1. Ларичев О.И. Теория и методы принятия решений: учебник для ВУЗов. М.: Университетская книга, Логос, 2006. 392 с.

2. Карпенко А.П., Митина Е.В., Семенихин А.С. Популяционные методы аппроксимации множества Парето в задаче многокритериальной оптимизации. Обзор // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2012. № 4. Режим доступа: <http://www.technomag.edu.ru/doc/363023.html> (дата обращения 01.10.2013).
3. Jong-hyun Ryu, Sujin Kim, Hong Wan. Pareto front approximation with adaptive sum method in multiobjective simulation optimization // Proc. of the 2009 Winter Simulation Conference (WSC), 2009, Austin, pp. 623-633. Available at: <http://www.informs-sim.org/wsc09papers/060.pdf> , accessed 01.10.2013.
4. Карпенко А.П., Савелов А.С., Семенихин А.С. Адаптивный метод взвешенных сумм в задаче Парето-аппроксимации // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2012. № 6. DOI: [10.7463/0612.0423283](https://doi.org/10.7463/0612.0423283)
5. Conn A.R, Gould N.I.M., Toint P.L. Trust-Region Methods. SIAM, Philadelphia, USA, 2000. 959 p. (MPS-SIAM Series on Optimization; no. 1).
6. Nocedal J., Wright S.J. Numerical Optimization. Springer, 1999. 634 p. DOI: 10.1007/b98874
7. Асатурян В.И. Теория планирования эксперимента. М.: Радио и связь, 1983. 248 с.
8. Соболев И.М., Статников Р.Б. Выбор оптимальных параметров в задачах со многими критериями: учеб. пособие для вузов. М.: Дрофа, 2006. 182 с.
9. Kim I.Y., de Weck O.L. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation // Structural and Multidisciplinary Optimization. 2005. Vol. 29, no. 2. P. 149-158. DOI: 10.1007/s00158-004-0465-1
10. Susan M. Sanchez, Paul J. Sanchez. Very large fractional and central composite design // ACM Transactions on Modeling and Computer Simulation. 2005. Vol. 15, no. 4. P. 362-377.
11. Хайкин С. Нейронные сети: полный курс : пер. с англ. М.: Издательский дом «Вильямс», 2006. 1104 с.

12. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // *Evolutionary Computation*. 2000. Vol. 8, no. 2. P. 173-195. DOI: 10.1162/106365600568202
13. Audet C., Savard G., Zghal W. Multiobjective optimization through a series of single-objective formulations // *SIAM Journal on Optimization*. 2008. Vol. 19, no. 1. P. 188-210. DOI: [10.1137/060677513](https://doi.org/10.1137/060677513)

Modified method of adaptive weighted sums in the problem of multi-criterion optimization

11, November 2013

DOI: 10.7463/1113.0632468

Karpenko A.P., Savelov A.S.

Bauman Moscow State Technical University, 105005, Moscow, Russian Federation

apkarpenko@mail.rus_savelov@mail.ru

This paper deals with the problem of multi-criterion optimization. It was assumed that the Pareto set is a solution to this problem. A promising approach to solving this problem is the adaptive weighted sum (AWS) method. Just like the classic weighted sum method based on the additive convolution of partial optimality criteria, the adaptive method also is based on the convolution. But the AWS method also proposes adaptation of weighting coefficients during iterations on the basis of the current location of the search subarea. This method works with meta-models of criterion functions in order to reduce computational costs. Results of our investigations showed that the AWS method provides a high quality Pareto-approximation in case of a convex, although, maybe, unconnected Pareto frontier. For problems with a concave Pareto frontier this method doesn't always provide sufficient quality of solution or it does but this takes too much time. In some cases the AWS method could provide an unacceptable solution caused by a special method of constraint satisfaction. The purpose of this work is to overcome specified disadvantages of the AWS method. A problem formulation for multi-criterion optimization was presented along with basic approaches to its solving. Several modifications of the ASW method proposed by the authors were considered. A brief description of the developed software implementing the method and its modifications were presented. Performance study of this software was carried out.

Publications with keywords: [multiobjective optimization](#), [Pareto set](#), [Pareto front](#), [adaptive weighted sum method](#)

Publications with words: [multiobjective optimization](#), [Pareto set](#), [Pareto front](#), [adaptive weighted sum method](#)

References

1. Larichev O.I. *Teoriia i metody priniatiia reshenii* [Theory and methods of decision making]. Moscow, Universitetskaia kniga, Logos, 2006. 392 p.

2. Karpenko A.P., Mitina E.V., Semenikhin A.S. Populyatsionnye metody approksimatsii mnozhestva Pareto v zadache mnogokriterial'noy optimizatsii. Obzor [Review: population methods of Pareto set approximation in multi-objective optimization problem]. *Nauka i obrazovanie MGTU im. N.E. Baumana* [Science and Education of the Bauman MSTU], 2012, no. 4. Available at: <http://www.technomag.edu.ru/doc/363023.html> , accessed 01.10.2013.
3. Ryu J.-H., Kim S., Wan H. Pareto front approximation with adaptive weighted sum method in multiobjective simulation optimization. *Proc. of the 2009 Winter Simulation Conference (WSC)*, 2009, Austin, pp. 623-633. Available at: <http://www.informs-sim.org/wsc09papers/060.pdf> , accessed 01.10.2013.
4. Karpenko A.P., Savelov A.S., Semenikhin A.S. Adaptivnyy metod vzveshennykh summ v zadache Pareto-approksimatsii [Adaptive weighted sum method for solving Pareto-approximation problem]. *Nauka i obrazovanie MGTU im. N.E. Baumana* [Science and Education of the Bauman MSTU], 2012, no. 6. DOI: [10.7463/0612.0423283](https://doi.org/10.7463/0612.0423283)
5. Conn A.R., Gould N.I.M., Toint P.L. *Trust-Region Methods*. SIAM, Philadelphia, USA, 2000. 959 p. (*MPS-SIAM Series on Optimization*; no. 1).
6. Nocedal J., Wright S.J. *Numerical Optimization*. Springer, 1999. 634 p. DOI: 10.1007/b98874
7. Asaturyan V.I. *Teoriya planirovaniya eksperimenta* [The theory of the design of experiments]. Moscow, Radio i svyaz', 1983. 248 p.
8. Sobol' I.M., Statnikov R.B. *Vybor optimal'nykh parametrov v zadachakh so mnogimi kriteriyami* [The choice of optimal parameters in problems with many criteria]. Moscow, Drofa, 2006. 182 p.
9. Kim I.Y., de Weck O.L. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 2005, vol. 29, no. 2, pp. 149-158. DOI: 10.1007/s00158-004-0465-1
10. Susan M. Sanchez, Paul J. Sanchez. Very large fractional and central composite design. *ACM Transactions on Modeling and Computer Simulation*, 2005, vol. 15, no. 4, pp. 362-377.
11. Haykin S. *Neural Networks: A Comprehensive Foundation*. 2nd ed. Prentice Hall, 1999. 823 p. (Russ. ed.: Haykin S. *Neironnye seti: polnyi kurs*. Moscow, Publishing House "Vil'iams", 2006. 1104 p.).
12. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 2000, vol. 8, no. 2, pp. 173-195. DOI: 10.1162/106365600568202
13. Audet C., Savard G., Zghal W. Multiobjective optimization through a series of single-objective formulations. *SIAM Journal on Optimization*, 2008, vol. 19, no. 1, pp. 188-210. DOI: [10.1137/060677513](https://doi.org/10.1137/060677513)