**M. M. Gourary,** Ph.D., e-mail: gourary@ippm.ru, **S. G. Rusakov,** D. Sc. profes., e-mail: rusakov@ippm.ru,
**S. L. Ulyanov,** D. Sc., e-mail: ulyas@ippm.ru, **M. M. Zharov,** Ph.D., e-mail: zarov@ippm.ru,
Institute for Design Problems in Microelectronics RAS

# The Iterative Method for Solving Parameterized System of Linear Equations with Multiple Right Hand Sides

*The problem of solving a system of linear equations with multiple right hand sides and matrix dependent on the parameter is considered. The iterative method with reuse of the previous results of matrix vector multiplications is suggested. The method provides the reduction of computational efforts and decrease of matrix-vector multiplications due to reuse of computed vectors. The proposed method is a generalization of known Krylov subspace algorithms. The application of proposed algorithms for periodic small-signal analysis of nonlinear electronic circuits is shown.*

***Keywords***: *system of linear algebraic equations, linear transform, orthogonalization, iterative methods, Krylov subspace, approximate solution*

**References**

1. **Saad Y.** *Iterative methods for Sparse Linear Systems.* Boston, PWS publishing company, 1996.
2. **Freund R. W.** A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems, *SIAM J. Sci. Stat. Comput.* 1993, vol. 14, no. 2, pp. 470—482.
3. **Saad Y., Schultz M.** GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. and Statist. Comput.*, 1986, vol. 7, no. 3, pp. 856—869.
4. **Gourary M. M., Rusakov S. G., Ulyanov S. L., Zharov M. M.** et al. Iterative Solution of Linear Systems in Harmonic Balance Analysis, *IEEE MTT-S Int. Microwave Symposium Digest*, 1997, pp. 1507—1510.
5. **Boyse W. E., Seidel A. A.** A block QMR method for computing multiple simultaneous solutions to complex symmetric systems, *SIAM J. Sci. Comput.*, 1996, vol. 17, no. 1, pp. 263—274.
6. **Freund R., Malhotra M.** A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides, *Linear Alg. Appl.*, 1997, vol. 254, no. 1—3, pp.197—257.
7. **Simoncini V., Gallopoulos E.** An iterative method for nonsymmetric systems with multiple right-hand sides, *SIAM J. Sci. Comput.*, 1995, vol. 16, no. 4, pp. 917—933.
8. **Simoncini V., Gallopoulos E.** A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides, *J. Comput. Appl. Math.*, 1996, vol. 66, no. 1, pp. 457—469.
9. **Baker A. H., Dennis J. M., Jessup E. R.** On improving linear solver performance: A block variant of GMRES, *SIAM J. Sci. Comput.*, 2006, vol. 27, no. 5, pp. 1608—1626.
10. **Simoncini V.** Restarted full orthogonalization method for shifted linear systems, *BIT Numer. Math.*, 2003, vol. 43, no. 2, pp. 459—466.
11. **Gu G-D., Simoncini V.** Numerical solution of parameter-dependent linear systems, *Linear Alg. Appl.*, 2005, vol. 12, no. 9, pp. 923—940.
12. **Baumann M., Gijzen M. B.** Nested Krylov methods for shifted linear systems, *SIAM J Sci Comput.*, 2015, vol. 37, no. 5, pp. S90—S112.
13. **Darnell D., Morgan R. B., Wilcox W.** Deflated GMRES for systems with multiple shifts and multiple right-hand sides, *Linear Algebra Appl.*, 2008, vol. 429, no. 10, pp. 2415—2434.
14. **Vergbitsky V. M.** *Chislennye metodi (mathematicheskiy analis i jbiknovennie differencialnye uravnenia)*, Moscow: Vysshaya shkola, 2001 (in Russian).
15. **Gourary M. M., Zharov M. M., Rusakov S. G., Ulyanov S. L.** Metod malosignalnogo analiza dla modelirovania mnogochastotnyh radiotechnicheskih shem, *Sb. trudov Vserossiyskoy nauchno-technicheskoy konferencii "Problemy razrabotki perspectivnyh mikroelectronnyh shem"*. Moscow: IPPM RAN, 2008, pp. 71—76 (in Russian).

УДК 519.6 DOI: 10.17587/it.24.370-386

**Т. А. Агасиев,** аспирант, e-mail: agtaleh@mail.ru,
**А. П. Карпенко,** д-р физ.-мат. наук, проф., зав. кафедрой, e-mail: apkarpenko@mail.ru,
Московский государственный технический университет им. Н. Э. Баумана

# Современные техники глобальной оптимизации. Обзор

*Представляем постановку задачи глобальной условной оптимизации, основной особенностью которой является высокая вычислительная сложность целевой функции. Даем определения таких сущностей, как характерные признаки задачи, базовые задача и алгоритм оптимизации, метазадача и метаалгоритм оптимизации, стратегия базовой задачи, индикатор эффективности стратегии. Определяем и приводим постановки мультииндикаторной, мультиклассовой и мультибюджетной задач метаоптимизации. На основе анализа около 100 публикаций даем обзор современных методов ландшафтного анализа целевых функций, а также методов метаоптимизации базовых алгоритмов. Наконец, представляем краткий обзор современного программного обеспечения, используемого для решения базовых и метазадач оптимизации.*

***Ключевые слова:*** *глобальная оптимизация, ландшафтный анализ, метаоптимизация, суррогатное моделирование, анализ эффективности алгоритмов оптимизации*

## Введение

Задачи глобальной оптимизации возникают во многих приложениях, например, при разработке комплексных инженерных изделий. Поэтому современные системы автоматизированного проектирования (*CAD*) и инженерного анализа (*CAE*) включают в себя программы,

реализующие, в том числе, алгоритмы непрерывной глобальной оптимизации.

Значительная вычислительная сложность моделей оптимизируемых объектов не позволяет осуществлять большое число обращений к этой модели. Поэтому в современном "промышленном" программном обеспечении задач оптимизации широко используют предварительные анализ и обработку этих задач, включая анализ исходных данных, понижение размерности пространства поиска, ландшафтный анализ целевой функции и т.д. [1]. По той же причине широко применяют предварительный выбор (с помощью методов метаоптимизации) наиболее эффективного алгоритма оптимизации, учитывающего особенности исходной оптимизационной задачи. Методологической основой этой техники является известная теорема о бесплатных завтраках (*No Free Lunch theorem* [2]), из которой следует, что если алгоритм $a_1$ эффективнее алгоритма $a_2$ при решении некоторой задачи, то обязательно найдется задача, для которой алгоритм $a_2$ окажется эффективнее алгоритма $a_1$.

Для подавляющего большинства практически значимых задач оптимизации неизвестны аналитические выражения целевой и ограничивающих функций. Поэтому общей практикой является применение прямых алгоритмов глобальной оптимизации, не использующих информацию о производных целевой функции [3]. Для достижения высокой эффективности в процессе решения сложных оптимизационных задач используют составные алгоритмы, комбинирующие различные "простые" алгоритмы оптимизации. Широко применяют, прежде всего, стохастические алгоритмы оптимизации: алгоритм симуляции отжига [4]; эволюционные, в том числе генетические алгоритмы [5, 6]; метаэвристические алгоритмы, вдохновленные природой, такие как алгоритм роя частиц [7], роя пчел [8], колонии муравьев [9] и т.д.

Для уменьшения числа обращений к целевой функции (числа *испытаний*) в процессе решения задач оптимизации, имеющих высокую вычислительную сложность, применяют аппроксимирующие модели целевой функции (*метамодели*, *суррогатные функции*). Примерами методов, использующих эту технику, являются *Efficient global optimization* (*EGO*) [10], *Sequential design for optimization* (*SDO*) [11], *SNOBFIT* [12].

Анализ эффективности алгоритмов оптимизации осуществляют путем экспериментального тестирования этих алгоритмов. Особенности проведения вычислительного эксперимента в целях оценки эффективности оптимизационных алгоритмов рассмотрены, например, в работе [13]. Для анализа эффективности алгоритмов оптимизации и их сравнения составлено большое число открытых наборов тестовых и инженерных задач. Значительная часть онлайн наборов таких задач собрана в коллекции Миттлмана (*Benchmarks for Optimization Software By Hans Mittelmann*) [14]. Наиболее полный обзор источников тестовых задач глобальной оптимизации выполнен в статье [15]. Набор *CUTE* (*Constrained and Unconstrained Testing Environment*) [16] содержит в формате *SIF* несколько сотен задач условной и безусловной оптимизации. Значительное число различных наборов тестовых задач и ссылок на соответствующую литературу собрано в [17]. Сложные инженерные задачи представлены в статье [18] и книге [19].

Широкий обзор существующих подходов к анализу результатов вычислительных экспериментов по исследованию эффективности оптимизационных алгоритмов выполнен в работе [20]. Для визуального сравнения результатов оценки эффективности алгоритмов оптимизации широкое распространение получили *профили производительности* (*Performance Profiles*) [21]. Авторы работы [22] рекомендуют использовать эту технику для попарного сравнения алгоритмов оптимизации в целях получения оценок их относительной эффективности. В работе [23] предложено использовать *профили данных* (*Data Profiles*) как дополнительное средство анализа эффективности алгоритмов оптимизации в условиях заданных ограничений на доступные вычислительные ресурсы.

Работа организована следующим образом. В п. 1 представляем постановку задачи и даем основные определения. В пп. 2, 3 рассматриваем две современные техники, используемые при "промышленном" решении задач глобальной оптимизации, — техники ландшафтного анализа целевой функции и метаоптимизации. В п. 3 представляем краткий обзор современного программного обеспечения, используемого для решения исходной задачи оптимизации и задачи метаоптимизации.

## 1. Постановка задачи и основные определения

***Постановка задачи.*** Рассматриваем задачу $q$ глобальной условной оптимизации

$$\min_{X \in D_X} f(X) = f(X^*) = f^*, \qquad (1)$$

где $X = (x_1, x_2, ..., x_{|X|})$ — $|X|$-мерный вектор варьируемых параметров (размерность задачи $q$); $D_X \subset \Re^{|X|}$ — область поиска; $f(X)$ — целевая функция; $X^*, f^*$ — искомые оптимальный вектор $X$ и значение целевой функции.

***Характерные признаки задачи.*** Результатом предварительной оценки свойств задачи $q = q(C)$ является $|C|$-мерный вектор *характерных признаков* (ХП) $C = (c_1, c_2, ..., c_{|C|})$ этой задачи. Различаем априорные и апостериорные ХП.

*Априорные* ХП прямо вытекают из постановки задачи (1) и включают в себя: размерность $|X|$ пространства варьируемых параметров; признак наличия или отсутствия ограничивающих функций в определении области $D_X$; тип целевой и ограничивающих функций; прогнозируемое число локальных экстремумов и т.д.

Авторы упомянутого выше набора *CUTE* тестовых задач оптимизации предложили универсальную систему обозначений задач оптимизации, основанную на использовании их априорных ХП [24]. Эта система каждой задаче $q(C)$ ставит в соответствие код вида

$$c_1c_2c_3c_4 - c_5c_6 - c_7 - c_8.$$

Здесь приняты следующие обозначения: $c_1$ определяет тип целевой функции (целевая функция не определена, является константной, линейной, квадратичной, представляет собой сумму квадратов, имеет общий вид); $c_2$ кодирует тип ограничений; $c_3$, $c_4$ представляют гладкость и степень дифференцируемости целевой функции; $c_5$ указывает на источник задачи (тестовая, имитирующая реальную задачу с отсутствием известного решения, практически значимая реальная задача оптимизации); $c_6$ свидетельствует о наличии или отсутствии дополнительных составных компонентов целевой функции, например вспомогательных функций в ее составе; величины $c_7 = |X|$, $c_8$ обозначают размерность задачи и число ограничивающих функций соответственно. Данную систему обозначений активно используют при тестировании алгоритмов оптимизации как позволяющую значительно упростить поиск и отбор необходимых тестовых задач.

*Апостериорные* ХП, в отличие от априорных, требуют вычислительных затрат на предварительные испытания целевой функции в области поиска $D_X$ в целях последующей экспертной и/или автоматической оценки результатов испытаний. Подчеркнем, что полученная при этом информация о результатах испытаний целевой функции может быть в последующем использована для решения задачи $q(C)$. К числу апостериорных ХП задачи оптимизации относят, прежде всего, ХП целевой функции.

Определение значений апостериорных ХП посредством экспертной оценки имеет ряд очевидных недостатков: низкая точность оценки, высокие временные затраты эксперта и т.д. Методы оценки значений апостериорных ХП целевой функции без использования экспертных оценок называют методами *ландшафтного анализа* (ЛА) [25].

***Базовая и метазадачи оптимизации.*** Современные алгоритмы оптимизации имеют, как правило, значительное число свободных параметров $(b_1, ..., b_{|B|}) = B$, от значений которых может существенно зависеть эффективность этих алгоритмов. Отсюда возникает задача *настройки алгоритма* — задача определения в каком-то смысле наилучших ("оптимальных") значений этих параметров. Вектор $B$ свободных параметров алгоритма оптимизации $a = a(B)$ называют *стратегией алгоритма a* (*Б-стратегией*). Множество $D_B$ допустимых Б-стратегий определяет набор алгоритмов $A(B) = \{a(B), B \in D_B\}$, т. е. *метод оптимизации* $A(B)$.

Выделяют два типа параметров в составе Б-стратегии — *числовые* и *категориальные.* Числовые параметры могут быть как непрерывными, так и дискретными. Категориальные параметры имеют относительно небольшое число допустимых значений, например, топологий соседства в алгоритме роя частиц [7].

Обычно при решении задачи $q(C)$ используют "умолчательные" стратегии алгоритма $a(B)$, полученные в результате проведенной ранее настройки этого алгоритма его авторами. Настройку алгоритма $a = a(B)$ может осуществлять также пользователь, который в этом случае должен иметь значительный опыт как в решении задач оптимизации, так и в использовании данного алгоритма оптимизации, должен понимать заложенные в алгоритм идеи, назначения его свободных параметров, знать детали программной реализации алгоритма и т.д.

Современным подходом к настройке алгоритмов оптимизации является подход на основе автоматизированного или автоматического решения задачи настройки как *задачи метаоптимизации* (*М-задачи).* Эта задача заключается в отыскании "оптимальной" стратегии $B^*$ алгоритма $a(B)$ при решении данной *базовой задачи оптимизации* $q(C)$ (*Б-задачи*) или *класса Б-задач оптимизации* $Q(D_C) = \{q(C)|C \in D_C\}$, где $D_C$ — множество допустимых значений компонентов вектора ХП. Настраиваемый алгоритм

оптимизации $a(B)$ называем *базовым алгоритмом* (*Б-алгоритмом*), а настраивающий — *метаалгоритмом* (*М-алгоритмом*).

Критерий эффективности Б-стратегий называем *индикатором эффективности* и обозначаем $e = e(q(C), a(B)) = e(C, B)$. М-задачу редко ставят и решают для одной Б-задачи, напротив, обычно М-задачу решают на некотором классе задач, который мы выше определили как $Q(D_C)$. Таким образом, *одноиндикаторную М-задачу* для класса Б-задач $Q(D_C)$ (*multi-instance problem tuning*) записываем в виде

$$\operatorname*{opt}_{B \in D_B} e(Q(D_C), B) = e(D_C, B^*). \qquad (2)$$

В зависимости от класса рассматриваемых М-алгоритмов искомой в задаче (2) является *статическая стратегия B** или *динамическая стратегия B*(t)*, которая программно или адаптивно изменяется в процессе решения Б-задачи (с ростом номера $t$ итераций Б-алгоритма).

Подчеркнем, что поскольку, как мы отмечали выше, в состав Б-стратегии могут входить как числовые, так и категориальные параметры, М-задача (2), в отличие от Б-задачи (1), представляет собой, вообще говоря, задачу смешанного программирования. В связи с этим при разработке М-алгоритмов принципиальной является используемая схема кодирования категориальных параметров Б-алгоритма [26].

В качестве индикатора эффективности Б-стратегий на классе задач $D_C$ обычно используют усредненные значения $e(D_C, B)$ индикатора $e(q(C), B)$ на этом классе задач, т. е. полагают, что $e(D_C, B) = \overline{e}(q(C), B)$, $C \in D_C$. Иногда рассматривают двумерный вектор индикаторов, первая компонента которого есть $\overline{e}(D_C, B)$, а вторая компонента — оценка стандартного отклонения $\sigma(D_C, B)$ индикатора $e(q(C), B)$ на том же классе задач. Другими словами, рассматривают М-задачу как мультизадачу М-оптимизации (см. ниже).

***Мультизадачи метаоптимизации.*** Выделяем три класса мультизадач М-оптимизации.

*Мультиндикаторные М-задачи.* В этих задачах эффективность Б-стратегий оценивают с помощью вектора индикаторов $E = (e_1, e_2, ..., e_{|E|}) = E(D_C, B)$. Аналогично задаче (2), мультииндикаторную М-задачу рассматриваем в постановке

$$\operatorname*{opt}^{\uparrow}_{B \in D_B} E(D_C, B) = E(D_C, \mathbf{B}^*), \qquad (3)$$

где индекс $\uparrow$ указывает на многокритериальность постановки задачи; $\mathbf{B}^*$ — набор "оптимальных" стратегий, принадлежащих множеству Парето задачи (3). В качестве компонентов вектора $E(D_C, B)$ аналогично задаче (2) могут использоваться их средние значения $\overline{e}_i(D_C, B)$, стандартные отклонения $\sigma_i(D_C, B)$; $i \in [1 : |E|]$.

По общим правилам решения задач многокритериальной оптимизации двумя основными методами решения задачи (3) являются метод скалярной сверки частных индикаторов и метод, предполагающий построение тем или иным способом конечномерной аппроксимации фронта и, тем самым, множества Парето этой задачи. Эти аппроксимации дают возможность ЛПР неформальными либо формализованными методами выбрать "оптимальную" с его точки зрения стратегию, изучить влияние различных компонентов Б-стратегии на эффективность Б-алгоритма, исследовать устойчивость значений индикаторов эффективности к вариациям стратегий и т.д. [27]. Для построения конечномерных аппроксимаций множества и фронта Парето задачи (3) используют как хорошо известные в многокритериальной оптимизации методы (*NSGA-II*, *SPEA-2* и т.д.), так и методы, специально разработанные для решения многоиндикаторной М-задачи.

*Мультиклассовые М-задачи* (*multu-problem tuning*). М-задачу (2) часто решают не на одном классе задач $Q(D_C)$, но на некотором наборе таких классов задач $Q(D_{C_1}), Q(D_{C_2}), \ldots, Q(D_{C_{|D|}})$, т. е. решают $|D|$ штук М-задач вида (2):

$$\operatorname*{opt}_{B \in D_B} e(D_{C_i}, B) = e(D_{C_i}, B_i^*), \quad D_{C_i} \in D_C, i \in [1 : |D|]. \qquad (4)$$

Относительно решений М-задачи (4) возникает вопрос: насколько чувствительна каждая из полученных "оптимальных" Б-стратегий $B_i^* = B^*(D_{C_i})$ к изменению (сужению или расширению) класса задач $Q(D_{C_i})$?

Ответ на этот вопрос отыскивают путем решения многокритериальной мультиклассовой М-задачи

$$\operatorname*{opt}^{\uparrow}_{B \in D_B} E(\mathbf{D}_C, B) = E(\mathbf{D}_C, \mathbf{B}^*). \qquad (5)$$

Здесь приняты следующие обозначения: $\mathbf{D}_C = \{D_{C_i} \in D_C, i \in [1 : |D|]\}$ — набор, определяющий рассматриваемые классы задач; $E(\mathbf{D}_C, B) =$

$= (e(D_{C_i}, B) = e_i(B), i \in [1 : |D|])$ — векторный индикатор эффективности.

Скалярной свертке индикаторов $e_i(B)$, $i \in [1 : |D|]$ трудно придать содержательный смысл. Поэтому используют построение конечномерной аппроксимации множества достижимости задачи (5) и/или отыскание таких же аппроксимаций ее множества и фронта Парето.

*Мультибюджетные М-задачи* возникают в связи с тем, что "оптимальная" Б-стратегия может сильно зависеть от используемой конфигурации вычислительной системы, т. е., вообще говоря, для разных доступных вычислительных ресурсов нужно отыскивать разные "оптимальные" стратегии. Применительно к каждой данной Б-задаче мощность этих ресурсов измеряют максимально допустимым числом испытаний целевой функции $f(X)$ — *бюджетом* задачи. При небольшом бюджете, обусловленном использованием ЭВМ малой мощности, "оптимальная" Б-стратегия обеспечивает преимущественную интенсификацию поиска. Напротив, мощные вычислительные системы позволяют использовать "оптимальные" стратегии, которые реализуют широко диверсифицированный поиск.

Поясним суть подхода к решению мультибюджетной М-задачи на примере решения одной Б-задачи $q(C)$. Положим, что рассматриваемый Б-алгоритм на каждой итерации решения этой задачи требует одинакового числа испытаний целевой функции $f(X)$. Тогда, очевидно, бюджет, выраженный в терминах числа испытаний $f(X)$, легко пересчитать в бюджет, определенный в терминах числа итераций Б-алгоритма $\hat{t}$. Обозначаем $\widehat{T} = (\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{|\widehat{T}|})$ интересующий исследователя набор бюджетов этого сорта.

Положим, что эффективность Б-стратегий оценивается с помощью одного индикатора эффективности $e(C, B)$, $B \in D_B$, подлежащего минимизации. Поскольку программное обеспечение глобальной оптимизации строят таким образом, чтобы в процессе итераций лучшие найденные решения Б-задачи не терялись, зависимость индикатора $e(qC, B)$ от текущего номера итерации $t$ представляет собой невозрастающую *функцию сходимости итерационного процесса*, т. е. для каждой из стратегий $B \in D_B$ имеет вид, представленный на рис. 1. Заметим, что фиксация этой зависимости требует пренебрежимо малых вычислительных затрат.

Рассмотрим некоторые допустимые Б-стратегии $B_1$, $B_2$, $B_3$. Рис. 1 показывает возможный

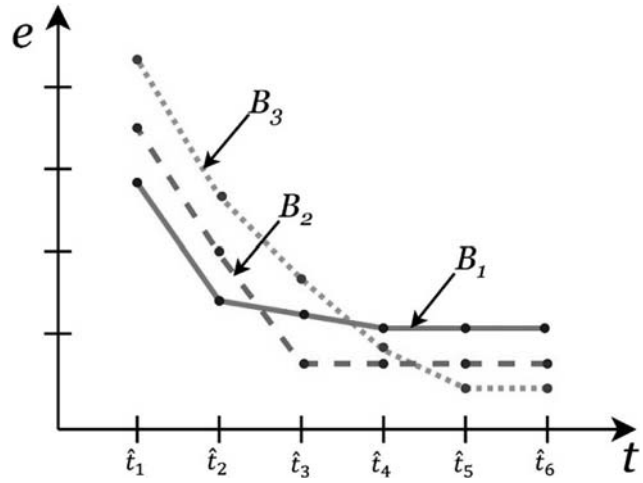


**Рис. 1. Зависимость лучшего найденного значения индиктора эффективности $e = e(C, B)$ от текущего номера $t$ итераций Б-алгоритма для некоторых допустимых стартегий $B_1$, $B_2$, $B_3$: $\hat{t}_1, \ldots, \hat{t}_6$ — бюджеты**

характер сходимости итерационных процессов, порожденных рассматриваемым Б-алгоритмом, для этих стратегий. Из рис. 1 следует, что "оптимальная" Б-стратегия $B^*$ для разных бюджетов имеет вид

$$B^* = \begin{cases} B_1, \hat{t} = \hat{t}_1, \hat{t}_2; \\ B_2, \hat{t} = \hat{t}_3, \hat{t}_4; \\ B_3, \hat{t} = \hat{t}_5, \hat{t}_6. \end{cases}$$

По общим правилам при решении мультибюджетной М-задачи на классе задача $Q(D_C)$ в качестве индикатора эффективности Б-стратегий можно использовать, например, среднее значение индикатора $e(C, B)$ на этом классе задач.

Заметим, что постановка М-задачи как мультибюджетной позволяет ввести новые метрики качества Б-стратегий по следующей схеме. Для каждого бюджета $\hat{t}_i$, $i \in [1 : |\widehat{T}|]$, ранжируем рассматриваемый набор Б-стратегий $\{B_j \in D_B\}$ по возрастанию индикатора $e(C, B)$, так что ранг стратегии $B_k$ равен $r(\hat{t}_i, B_k) = 1$, если для бюджета $\hat{t}_i$ индикатор $e(C, B_k)$ имеет минимальное значение; аналогично $r(\hat{t}_i, B_l) = 2$, где $B_l \in \{B_j\} \backslash B_k$ и т.д. Каждую из стратегий набора $\{B_j\}$ характеризуем числом бюджетов $\hat{t}_i$, $i \in [1 : |\widehat{T}|]$, при которых она является лучшей, т. е. имеет ранг $r = 1$. Мерой качества стратегии $B_j$ может служить также площадь под непрерывным графиком $e(C, B_j, \hat{t})$ (меньшая площадь означает более быструю сходимость соответствующего итерационного процесса и более высокое качество стратегии).

Как и в одноиндикаторных М-задачах, искомой в рассмотренных мультизадачах М-оптимизации могут является статическая $B^*$ или динамическая $B^*(t)$ стратегии.

Введем в заключение этого раздела следующие обозначения: $1M^2$ — одна из рассмотренных мультизадач М-оптимизации; $2M^2$ — комбинированная мультизадача М-оптимизации, объединяющая любые две из этих мультизадач; $3M^2$ — аналогичная задача, включающая в себя все три мультизадачи.

## 2. Методы ландшафтного анализа целевой функции

Для определения значений апостериорных ХП целевой функции методами ЛА необходима *обучающая выборка* — набор точек $X \in D_X$ и соответствующих значений целевой функции $f(X)$. Найденные значения ХП определяются не только ландшафтом функции $f(X)$, но и методом планирования эксперимента, использованным при формировании выборки [6]. Потому предпочтение отдают адаптивным методам планирования эксперимента [28], которые по сравнению с неадаптивными методами позволяют получить более точные значения ХП целевой функции за меньшее число испытаний.

*Классический метод ландшафтного анализа* [25] предлагает в общей сложности 50 числовых признаков ландшафта функции $f(X)$, сгруппированных в шесть так называемых свойств: выпуклость, степень кривизны, $y$-распределение, ярусность, мультимодальность, метамодельные свойства. Здесь под $y$-распределением понимают статистические оценки плотности вероятности распределения значений $f(X)$ в точках обучающей выборки, число экстремумов функции плотности вероятности, степень близости этой функции к нормальному закону распределения. Ярусность функции $f(X)$ оценивают как вероятность того, что ее значения превышают заданное значение. Метамодельные признаки $f(X)$ вычисляют на основании результатов оценки точности ее линейных или квадратичных регрессионных М-моделей, построенных на используемой обучающей выборке. Подчеркнем, что определение некоторых из указанных свойств, например свойства мультимодальности, требует проведения дополнительных испытаний функции $f(X)$ в новых точках области $D_X$, что накладывает сильные ограничения на применение данного метода ЛА для исследования задач, имеющих высокую вычислительную сложность целевой функции.

*Метод клеточного отображения* (*Cell Mapping*) [29]. В этом случае область поиска $D_X$ по каждому из измерений пространства $\Re^{|X|}$ равномерно разбивают на подобласти, называемые *клетками*. Для каждой клетки вычисляют ХП функции $f(X)$, оценивающие выпуклость, однородность градиента функции и угол между векторами, которые соединяют центр клетки с точками выборки внутри нее, имеющими наибольшее и наименьшее значение $f(X)$ (рис. 2). В случае, если функция $f(X)$ имеет малое число локальных оптимумов, суммарный разброс вычисленных значений признаков по всем клеткам является небольшим, поскольку соответствующие точки находятся на границах этих клеток.

*Метод обобщенного клеточного отображения* (*Generalized Cell Mapping*) [29]. Область $D_X$ аналогичным образом разбивают на клетки. Каждой клетке ставят в соответствие одно значение функции $f(X)$, например, в точке $X$, ближайшей к центру клетки. ХП ландшафта функции $f(X)$ вычисляют на основе оценки вероятностей перехода между клетками с помощью поглощающих цепей Маркова.

*Метод барьерных деревьев.* После дискретизации области поиска $D_X$ методом обобщенного клеточного отображения применяют *барьерные деревья (Barrier Trees)* [30]. Идея заключается в представлении полученного множества клеток

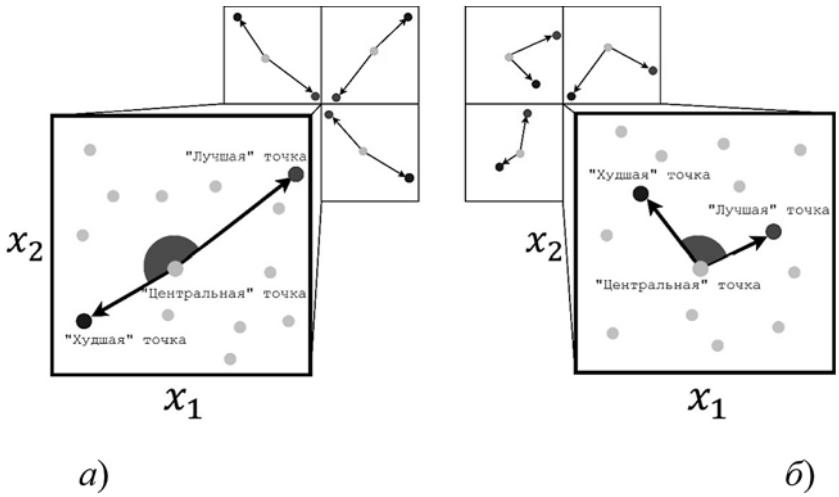


**Рис. 2. Метод клеточного отображения: характерные точки клетки для унимодальной (*а*) и мультимодальной (*б*) функций; $|X| = 2$**

в виде дерева, в котором клетки, содержащие локально "оптимальные" точки целевой функции $f(X)$, соответствуют листьям. Значения ХП функции $f(X)$ определяют на основе оценок различных параметров построенного дерева: число листьев; высота дерева; статистические оценки разности значений $f(X)$ в клетках, соответствующих соседним узлам дерева, и т.д.

*Метод информационного содержания* (*Information Content*) [31] позволяет оценить численные характеристики зашумленности и мультимодальности ландшафта целевой функции $f(X)$. Сначала точки $X_i$, $i \in [1 : l]$, обучающей выборки упорядочивают тем или иным образом и вычисляют перепад значений $f(X)$ между соседними точками выборки. Затем последовательность значений $f(X_i) = f_i$ преобразуют в символьный набор $S(\varepsilon) = \{s_1, ..., s_{l-1}\}$ по правилу

$$s_i = \begin{cases} \bar{1}, \delta_i < -\varepsilon; \\ 0, |\delta_i| \leqslant \varepsilon; \\ 1, \delta_i > \varepsilon, \end{cases} \quad \delta_i = \frac{f_{i+1} - f_i}{\|X_{i+1} - X_i\|}, \quad i \in [1 : l-1],$$

где $\varepsilon$ — параметр, определяющий чувствительность метода; $\|\cdot\|$ — евклидова норма. Полученную последовательность символов используют для вычисления искомых ХП на основании значений *функции информационного содержания*

$$ic(\varepsilon) = -\sum_{a \neq b} p_{ab}(\varepsilon) \log_6 p_{ab}(\varepsilon), \; a, b \in \{\bar{1}, 0, 1\},$$

где $ab$ — блок последовательных символов из набора $S(\varepsilon)$, например $\bar{1}1$, 01, 11; $p_{ab}$ — вероятность обнаружения блока $ab$ в наборе $S(\varepsilon)$.

Известно некоторое число других методов ЛА. Например, метод *Nearest-Better Clustering* [32, 33] позволяет идентифицировать единичные большие скачки мультимодальной целевой функции $f(X)$. Метод главных компонент [34] определяет ландшафт числом главных компонент рассматриваемой выборки, при котором значение остаточной (объяснительной) дисперсии не превышает заданного значения. Предложены также ХП на основе оценки дисперсии попарных расстояний между всеми точками выборки и лучшими из этих точек [14].

## 3. Метаоптимизация

### *3.1. Общие сведения*

В соответствии с классификацией, предложенной в работе [35], на верхнем уровне иерархии выделяют две группы методов М-оптимизации — методы *настройки* параметров и методы *управления* параметрами (рис. 3, см. вторую сторону обложки). При настройке параметров выбранную стратегию не меняют в ходе решения Б-задачи. Информацию об эффективности Б-алгоритма с данной стратегией получают после окончания решения этой задачи. В случае управления параметрами стратегию варьируют в процессе решения Б-задачи. Детальное сравнение этих групп методов М-оптимизации выполнено в работе [36].

Методы настройки параметров являются более универсальными в сравнении с методами управления параметрами, поскольку не требуют доработки Б-алгоритма, который в этом случае рассматривается как "черный ящик". Все пользователи Б-алгоритма после окончания процесса настройки получают "оптимальную" стратегию для решения определенного класса Б-задач, что избавляет их от дальнейших вычислительных затрат на М-оптимизацию. Методы настройки позволяют дополнять накопленный опыт решения Б-задач, уточняя найденные "оптимальные" Б-стратегии.

Выделяют два основных подхода к настройке параметров [37]. Целью первого подхода является быстрое отыскание "оптимальной" Б-стратегии путем *интенсификации* поиска в пространстве стратегий. Цель второго подхода состоит в получении информации о поведении Б-алгоритма в процессе решения им Б-задач для более глубокого понимания особенностей функционирования этого алгоритма. Достижение последней цели требует обширного исследования пространства Б-стратегий, т. е. *диверсификации* поиска.

В работе [37] выделены следующие основные подходы к "одноиндикаторной" настройке параметров (рис. 4, см. вторую сторону обложки), отличающиеся, прежде всего, соотношением интенсификационной и диверсификационной составляющих поиска "оптимальной" Б-стратегии:

- *стохастические поисковые методы*, "заточенные" под быстрое отыскание "оптимальных" стратегий;
- *методы сэмплирования (sampling)*, позволяющие провести обширное исследование пространства стратегий;
- *скрининговые методы*, содержащие различные техники раннего выявления перспективных стратегий.
- *метамодельные методы*, использующие суррогатную модель индикатора эффективности стратегий (*М-модель*) и ориентированные на интенсификацию поиска "оптимальных" стратегий;

- *комбинированные методы.*

Выделяем также следующие классы методов настройки параметров:

- методы, ориентированные на решение мультизадач М-оптимизации;
- методы настройки параметров для параллельных вычислительных систем.

В терминах [37] различают *итеративные* и *не итеративные* методы настройки. В случае не итеративной настройки тем или иным образом генерируют набор исследуемых стратегий, из числа которых и выбирают "оптимальную" для решения рассматриваемого класса Б-задач. При итеративной настройке параметров (штриховые стрелки на рис. 4) набор Б-стратегий, из числа которых выбирают "оптимальную", не фиксирован. Новые стратегии генерируют на основе информации, поступающей в процессе настройки.

Методы однократной настройки могут быть как итеративными, так и не итеративными, в то время как методы перманентной настройки являются строго итеративными. Наоборот, итеративные методы можно использовать как для однократной, так и для перманентной настройки Б-алгоритмов. Не итеративные методы ориентированы на однократную настройку.

В пп. 3.2—3.6 последовательно рассматриваем представленные выше "одноиндикаторные" методы настройки, а в пп. 3.7, 3.8 — методы настройки для решения мультизадач М-оптимизации и параллельные методы настройки.

### *3.2. Стохастические поисковые методы*

Исторически первым представителем стохастических методов М-оптимизации является метаэволюционный алгоритм (*meta-EA*), предложенный в 1978 г. [38]. Параметры в этом методе настраиваются с помощью эволюционного алгоритма. Широкое исследование эффективности метода не было выполнено в силу его относительно высокой вычислительной сложности. Обширные исследования были выполнены с использованием близкого метагенетического алгоритма (*meta-GA*). Здесь стратегиям Б-алгоритма поставлены в соответствие особи М-алгоритма, а индикатору эффективности стратегий — фитнес-функции М-алгоритма. Заметим, что такая интерпретация М-задачи позволяет использовать для настройки параметров любой эволюционный алгоритм. Исследования показали высокую эффективность данного М-алгоритма настройки параметров [39].

Метод *FocusedILS* [40] использует в качестве М-алгоритма гибридизацию алгоритмов *эволюционной стратегии* и локальной оптимизации. Авторы метода предложили специальную технику для сравнения эффективности испытываемых и выявления "оптимальных" Б-стратегий, позволяющую уменьшить общее число испытаний стратегий (обращений к Б-алгоритму). Похожее решение предложено в работе [41], авторы которой добавили скрининговый метод *гонки* (см. п. 3.4). Недостатком подобных гибридизаций является усложнение М-алгоритма и, как следствие, рост вычислительных затрат на М-оптимизацию.

Метод *SMAC* (*Sequential Model-based Algorithm Configuration*) [42] представляет собой модифицированный метод *FocusedILS*. Основной особенностью *SMAC* является использование М-модели фитнесс-функции М-задачи для локального уточняющего поиска "оптимальной" Б-стратегии. Перспективность стратегии определяют на основе модели случайного леса (*Random Forest Model*) [43].

### *3.3. Методы сэмплирования*

Подход к М-оптимизации на основе сэмплирования ориентирован на сокращение общего числа испытаний стратегий (в ущерб, естественно, широте поиска). Методы данного класса чаще используют на стадии инициализации поиска "оптимальной" стратегии с помощью, например, метамодельных методов, нежели в качестве самостоятельных М-методов. Наиболее известны следующие не итеративные методы, реализующие данный подход: метод на основе *латинского квадрата* (*Latin-Square*) [44]; метод на основе *алгоритма Тагучи* (*Taguchi Orthogonal Arrays*) [45]. Известными примерами итеративных методов сэмплирования являются методы *CALIBRA* [46] и *Empirical Modeling of Genetic Algorithms* [44].

В работе [47] предложены следующие простые итеративные методы настройки, использующие сэмплирование: метод на основе паттернов (*Pattern Search*, *PS*); метод локальной унимодальной выборки (*Local Unimodal Sampling*, *LUS*). Результаты проведенного авторами исследования не показали заметного снижения эффективности "оптимальных" Б-стратегий, найденных этими методами, по сравнению со стратегиями, полученными с помощью некоторых других значительно более ресурсоемких М-алгоритмов.

### *3.4. Скрининговые методы*

Целью скрининговых методов М-оптимизации является выявление "оптимальной" стратегии за минимальное число запусков Б-алгоритма. Скрининговые методы сформировались под влиянием методов ранжирования и выбора лучшей из имеющихся имитационных моделей исследуемой системы по степени достоверности и точности результатов моделирования [48]. Б-стратегии в этом случае ставится в соответствие имитационная модель, а решению Б-задачи — испытание этой модели. М-алгоритм соответствует методу отбора имитационных моделей.

В работе [49] выполнено сравнение эффективности наиболее известных скрининговых методов: *Interactive Analysis* [50]; *Ranking and Selection* [51]; *Multiple Comparison Procedures* [52]; *Fully Sequential Indifference-zone Selection procedure* [53]. Показано, что указанные методы значительно отличаются необходимым числом испытаний исследуемых моделей и гарантиями отбора лучшей из них.

Скрининговые методы основаны на допущении, что результаты стохастической имитации распределены по нормальному закону. Эти допущения являются слабо обоснованными, но могут быть в значительной мере удовлетворены путем перегруппирования результатов стохастических испытаний [49]. Преимуществом скрининговых методов можно считать определенные гарантии локализации "оптимальной" Б-стратегии. Уровень доверия к "оптимальности" полученной стратегии можно оценить, например, с помощью метода *Multiple Comparison Procedures* [48, 49].

Известным представителем итеративных скрининговых методов является метод *I/F-RACE* [54], построенный на основе метода *F-RACE* [60] и сочетающий в себе техники скрининга и локального уточняющего поиска. Область допустимых стратегий $D_B$ в этом методе покрывают сеткой с относительно небольшим числом узлов. С помощью *гонок* [55] отбирают наиболее перспективные Б-стратегии, на основании которых строят аппроксимацию функции плотности вероятности локализации "оптимальных" стратегий в подобластях множества $D_B$. Полученную вероятностную М-модель используют для генерации новых Б-стратегий.

Применительно к настройке параметров часто используют подход на основе *гонок*, который впервые был предложен в методе *Hoeffding Races* [55]. В сущности, этот подход незначительно отличается от рассмотренных выше методов, но позволяет исследовать большее число Б-стратегий. В основе подхода лежит правило изменения числа запусков Б-алгоритма с каждой из стратегий (числа испытаний стратегии) для оценки ее эффективности. Начинают гонки с малого числа испытаний, по результатам которых отбрасывают явно неэффективные стратегии. В последующих итерациях гонок постепенно увеличивают число испытываемых стратегий для более точной оценки их эффективности. В результате более перспективные стратегии исследуются чаще, чем малоэффективные. Преимущество гонок состоит в том, что они не требуют каких-либо допущений о распределении значений используемого индикатора эффективности Б-алгоритма (см. методы *Hoeffding Races* [55], *F-RACE* [56]).

### *3.5. Метамодельные методы настройки*

Данный класс М-методов настройки основан на использовании аппроксимирующей (суррогатной) модели (М-модели) индикатора эффективности $e(C, B)$, построение которой имеет целью уменьшить число вычислительно емких испытаний Б-стратегий. Аналогично методу ЛА при использовании этого класса методов настройки качество предсказаний М-модели напрямую зависит от метода планирования эксперимента, с помощью которого осуществляется генерация соответствующей обучающей выборки [57].

Часто для построения М-модели используют метод регрессии [58, 59]. Точность модели, а значит, и качество найденных "оптимальных" Б-стратегий оказываются в этом случае относительно низкими. Итеративные метамодельные методы позволяют и при использовании регрессионной М-модели найти приемлемые "оптимальные" Б-стратегии.

Одним из наиболее известных метамодельных методов настройки является *процедура Коя* (*Coy's procedure*) [60]. Основная идея процедуры состоит в использовании методов локального поиска для уточнения Б-стратегий, полученных с помощью М-модели. На первом этапе область $D_B$ дискретизируют и в узлах полученной сетки вычисляют значения индикатора эффективности $e(C, B)$. На основании полученных данных строят линейную регрессионную М-модель этого индикатора. На втором этапе с помощью построенной модели генерируют и испытывают Б-стратегии с использованием метода наискорейшего спуска.

*Метод SPO* (*Sequential Parameter Optimization*), в отличие от двухступенчатой процедуры Коя, постоянно обновляет М-модель [61, 62]. Каждая итерация М-алгоритма в этом случае включает в себя следующие действия: обновление М-модели на основе результатов предыдущей итерации; генерация набора Б-стратегий и прогнозирование их эффективности с использованием текущей М-модели; испытания наиболее перспективных стратегий.

Качество "оптимальных" Б-стратегий, найденных метамодельными методами настройки, во многом зависит от типа используемой М-модели. Исследование эффективности различных типов этих моделей (суррогатных функций), способов их построения и использования в целях настройки параметров Б-алгоритма выполнено в работе [63].

### *3.6. Комбинированные методы*

*Метод SEPaT* (*Simple Evolutionary Parameter Tuning*) [26] первоначально позиционировался авторами как стохастический М-метод. Однако используемый методом *SEPaT* подход к оценке эффективности Б-стратегий может применяться в любых других одноиндикаторных М-методах. Метод *SEPaT* реализует различные правила быстрого отбора эффективных Б-стратегий, в том числе стратегий, содержащих категориальные параметры. Эффективность стратегий вычисляют на основе среднего и стандартного отклонений индикатора эффективности $e(D_C, B)$. Полагается, что одна Б-стратегия лучше другой, если она эффективнее в среднем и имеет меньшее стандартное отклонение значений индикатора эффективности. В противном случае, лучшую стратегию выбирают на основе результатов теста Велча (*Welch's t-test* [64]), т. е. на основе оценки величины

$$e'(q_i, B_1, B_2) = \sum_{i=1}^{n} \frac{\bar{e}(q_i, B_1) - \bar{e}(q_i, B_2)}{\sqrt{\frac{\sigma^2(q_i, B_1) - \sigma^2(q_i, B_2)}{m}}},$$

$$q_i = q_i(C) \in Q(D_C),$$

где $B_1$, $B_2$ — сравниваемые допустимые стратегии; $\bar{e}(q_i, B)$, $\sigma(q_i, B)$ — среднее значение и стандартное отклонение индикатора эффективности $e(q, B)$, полученные на основе решения $m$ раз Б-задачи $q_i$. В случае минимизации индикатора эффективности $e(q, B)$ стратегия $B_1$ лучше стратегии $B_2$, если $e'(q, B_1, B_2) < 0$. Если необходим анализ эффективности более двух стратегий, то их сравнивают попарно каждую с каждой, и стратегию с наименьшим числом поражений признают лучшей, а в случае равенства числа проигрышей сравнивают число выигрышей. Таким образом, найденная "оптимальная" стратегия оказывается наиболее эффективной в среднем для всех рассматриваемых Б-задач класса $D_C$. Более детально метод рассмотрен в работе [65], где его сравнивают с методами гонок [55] и *ParamILS* [66].

### *3.7. Методы решения мультизадач М-оптимизации*

*Метод M-FETA* (*Multi-Function Evolutionary Tuning Algorithm*) [27] использует специализированный алгоритм мультииндикаторной настройки параметров. Основным преимуществом алгоритма является наличие специальных операторов, понижающих число требуемых запусков Б-алгоритма для исследования каждой из испытываемых Б-стратегий. С этой целью эффективность некоторых стратегий оценивают с помощью значений индикаторов эффективности "соседних" стратегий; в перспективных подобластях множества $D_B$ генерируют большее число стратегий, что уменьшает расстояние между "соседними" перспективными стратегиями и позволяет уточнить оценки их эффективности.

*Метод REVAC* (*Relevance Estimation and Value Calibration of Parameters*) в исходном варианте относится к эвристическим итеративным стохастическим поисковым методам настройки [67], однако известна его мультииндикаторная модификация. В этой модификации фронт Парето Б-стратегий не строят, но используют оригинальную методику многоиндикаторного ранжирования стратегий на основании оценки их эффективности при решении различных Б-задач. Метод позволяет использовать гонки [55] в целях уменьшения числа испытаний Б-стратегий.

*Метод Bonesa* [68] в своей основе является итеративным методом настройки параметров, использующим М-модель. Для уменьшения числа запусков стохастических Б-алгоритмов с каждой из испытываемых Б-стратегий используется фильтр Гаусса [69], который уменьшает уровень зашумленности индикатора эффективности. Сравнение эффективности стратегий выполняют с помощью гонок [55]. Для построения фронта Парето метод *Bonesa* использует алгоритм *SPEA2* [70].

*Метод EMOPaT* (*Evolutionary Multi-Objective*) [26] является мультиклассовой модификацией

метода *SEPaT*. В качестве М-алгоритма использован алгоритм *NSGA-II* [71]. Метод включает в себя анализ Парето-оптимальных решений с помощью простого и эффективного подхода *innovation (innovation through optimization)* [72]. Компонентами вектора Б-стратегий в методе *EMOPaT* могут быть категориальные параметры. Схема кодирования этих параметров представлена в работе [26].

*Метод Flexible Budget* [73] относится к классу мультибюджетных методов М-оптимизации. Результатом настройки является набор Б-стратегий, эффективных при различных бюджетах.

### *3.8. Методы для параллельных вычислительных систем*

*Метод ReACT* (*Real-time Algorithm Configuration through Tournaments*) [74] позволяет одновременно испытывать несколько Б-стратегий посредством гонок с использованием многопроцессорных систем. На каждом из доступных процессоров запускают экземпляр Б-алгоритма с одной из допустимых стратегий. После завершения всех запущенных Б-алгоритмов определяют стратегию-победителя. Если по истечении заданного ограничения по времени ни один из Б-алгоритмов не завершится, то победителем считают стратегию с лучшим достигнутым значением целевой функции $f(X)$. Как и *SEPaT*, данный метод может быть скомбинирован с любыми методами одноиндикаторной настройки.

## 4. Современное программное обеспечение глобальной оптимизации

### *4.1. Базовая оптимизация*

Значительное число методов глобальной оптимизации, не использующих аналитические или численные производные в процессе поиска, имеют открытые реализации: *Bound Optimization BY Quadratic Approximation* (*BOBYQA*) [75]; *Design Analysis Kit for Optimization and Terascale Applications* (*DAKOTA*) [76]; *GLOBAL* [77]; *Hybrid Optimization Parallel Search PACKage* (*HOPSPACK*) [78]; *NEWUOA* [79]; *NOMAD* [80]; *SID-PSM* [81] и т.д.

Для решения Б-задач, имеющих высокую вычислительную сложность, разработаны специальные программные комплексы, представленные ниже.

*Платформа modeFRONTIER*, помимо традиционных инструментов для оптимизации, включает в себя большой набор гибридных алгоритмов [82], единственным свободным параметром которых является максимальное число испытаний целевой функции. Алгоритм *FAST* использует *RSM* модели для исследования наиболее перспективных областей пространства поиска. Алгоритм *HYBRID* сочетает устойчивость генетических алгоритмов и точность градиентных. В рамках алгоритма *SAnGeA* реализован подход, который позволяет определить наиболее значимые компоненты вектора варьируемых параметров $X$. Алгоритм *pilOPT* комбинирует методы локального, глобального поиска и метамоделирования.

*Программный продукт HEEDS Professional* реализует проприетарный гибридный адаптивный алгоритм *SHERPA* [83]. Алгоритм использует несколько методов локальной и глобальной оптимизации, автоматически модифицируя их параметры по мере поступления информации о ландшафте целевой функции $f(X)$. Единственным свободным параметром алгоритма является максимально допустимое число испытаний $f(X)$.

*Модуль Generic Tool for Optimization* (*GTOpt*) [84], поставляемый компанией *Datadvance*, спроектирован для решения сложных прикладных инженерных оптимизационных задач. Модуль содержит оригинальные высокоэффективные алгоритмы одно- и многокритериальной оптимизации, использующие метамодели целевых функций

*Программный комплекс IOSO NM* реализует алгоритм *Indirect Optimization on the basis of Self-Organization* (*IOSO*) [85], разработанный компанией *Сигма Технология*. Программный комплекс пригоден для решения широкого спектра оптимизационных задач, в том числе с одновременным использованием нескольких моделей оптимизируемого объекта. Алгоритм *IOSO* использует эволюционные алгоритмы с самоорганизацией. Высокая эффективность алгоритма обеспечивается набором встроенных правил адаптивного выбора алгоритма оптимизации.

### *4.2. Программное обеспечение для настройки параметров*

*Универсальный программный продукт SMAC* реализует одноименный метод настройки параметров Б-алгоритмов, а также иных параметризованных объектов, например имитационных моделей, алгоритмов машинного обучения

и т.д. [86]. Продукт *SMAC* не имеет графического интерфейса, запуск М-алгоритма приходится проводить с помощью командной строки. Пользователь должен предоставить программе бинарный файл Б-алгоритма, конфигурационные файлы с описанием стратегий и индикатора эффективности Б-алгоритма. Результаты настройки также выдаются в виде файла.

Подобный способ взаимодействия с пользователем реализован и в *программном комплексе ParamILS* [87]. Основные различия между *SMAC* и *ParamILS* заключаются в следующем.

- Комплекс *ParamILS* позволяет использовать только категориальные свободные параметры Б-алгоритма, в то время как *SMAC* поддерживает и числовые параметры. Также *SMAC* предоставляет более широкий выбор индикаторов эффективности Б-стратегий.
- Программа *SMAC* имеет возможность гибкой настройки временных и вычислительных ограничений М-алгоритма и позволяет использовать дополнительную информацию о ХП исследуемых Б-задач.

*Программный комплекс DGGA Algorithm Configurator* реализует метод настройки *GGA* [88], поддерживающий как категориальные, так и числовые параметры Б-алгоритма. Метод *GGA* использует технику гонок для поиска "оптимальных" Б-стратегий. Комплекс ориентирован на решение М-задач с применением параллельных вычислительных систем. Аналогично программным комплексам *SMAC*, *ParamILS*, для конфигурирования М-алгоритма и описания исходных данных М-задачи используются файлы специальной структуры.

*Программный пакет irace* [89] разработан на основе расширенного итеративного метода настройки *I/F-Race* [54], использующего гонки. Пакет предоставляет возможность параллельного исполнения М-алгоритма, задания начального набора стратегий, указания признаков "плохих" стратегий, которые должны быть исключены из процедуры поиска. Пакет широко применяют для настройки Б-алгоритмов непрерывной и дискретной оптимизации [90], в том числе алгоритмов многокритериальной оптимизации [91].

## Заключение

Современное состояние техник глобальной оптимизации позволяет сделать следующие выводы.

В целях сокращения вычислительных затрат в программном обеспечении "промышленной" глобальной оптимизации возрастает поддержка предварительного анализа постановки задачи на основе использования известных и новых алгоритмов анализа данных, понижения размерности пространства поиска, ландшафтного анализа целевой функции.

Имеет место тенденция преимущественного использования гибридных адаптивных алгоритмов оптимизации, объединяющих несколько стохастических алгоритмов глобальной оптимизации и алгоритмов локальной оптимизации (мультимемеевые алгоритмы).

Повышается "интеллектуальность" программных комплексов глобальной оптимизации, в том числе за счет реализации в них методов метаоптимизации базовых алгоритмов оптимизации. Если в настоящее время, как правило, используются методы однократной настройки параметров этих алгоритмов, то перспективным является использование методов перманентной настройки, а также адаптивных и самоадаптивных методов управления параметрами. Вместе с тем сейчас, почти без исключений, мультизадачи метаоптимизации решают только по отдельности, т. е. как $1М^2$-задачи. Актуальной является проблема совместного решения таких задач (т. е. решения $2М^2$- и $3М^2$-задач). При этом возникают сложные задачи многокритериального принятия решений, требующие одновременного анализа более одного многомерного фронта (множества) Парето. В целом современные методы многокритериального принятия решений еще недостаточно широко используются в процессе решения задач метаоптимизации. Например, перспективным, на наш взгляд, является использование подходов, основанных на выявлении так называемой функции предпочтений ЛПР [92, 93].

Для решения современных практически важных задач оптимизации все более широко используют параллельные вычислительные системы, имеющие разные архитектуры — вычислительные кластеры, системы с общей памятью, системы на основе графических процессорных устройств, слабосвязанные вычислительные системы типа ГРИД-систем и т.д. В связи с этим актуальными являются проблемы разработки методов, алгоритмов и соответствующего программного обеспечения для решения базовых задач оптимизации и задач метаоптимизации, ориентированных на указанные классы параллельных систем.

**Список литературы**


1. **Shan S., Wang G. G.** Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions // Structural and Multidisciplinary Optimization. 2010. Vol. 41, N. 2. P. 219—241.

2. **Wolpert D. H., Macready W. G.** No free lunch theorems for optimization // IEEE transactions on evolutionary computation. 1997. Vol. 1, N. 1. P. 67—82.

3. **Rios L. M., Sahinidis N. V.** Derivative-free optimization: a review of algorithms and comparison of software implementations // Journal of Global Optimization. 2013. Vol. 56, N. 3. P. 1247—1293.

4. **Kirkpatrick S.** Optimization by simulated annealing: Quantitative studies // Journal of statistical physics. 1984. Vol. 34, N. 5—6. P. 975—986.

5. **Liepins G. E., Hilliard M. R.** Genetic algorithms: Foundations and applications // Annals of operations research. 1989. Vol. 21, N. 1. P. 31—58.

6. **Hansen N.** The CMA evolution strategy: a comparing review // Towards a new evolutionary computation. Springer, Berlin Heidelberg, 2006. P. 75—102.

7. **Poli R., Kennedy J., Blackwell T.** Particle swarm optimization // Swarm intelligence. 2007. Vol. 1, N. 1. P. 33—57.

8. **Karaboga D., Basturk B.** A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm // Journal of global optimization. 2007. Vol. 39, N. 3. P. 459—471.

9. **Dorigo M., Birattari M., Stutzle T.** Ant colony optimization // IEEE computational intelligence magazine. 2006. Vol. 1, N. 4. P. 28—39.

10. **Jones D. R., Schonlau M., Welch W. J.** Efficient global optimization of expensive black-box functions // Journal of Global optimization. 1998. Vol. 13, N. 4. P. 455—492.

11. **Cox D. D., John S.** SDO: A Statistical Method for Global Optimization // Multidisciplinary Design Optimization: State of the Art. 1997. Vol. 80. P. 315—329.

12. **Huyer W., Neumaier A.** SNOBFIT — Stable noisy optimization by branch and fit // ACM Transactions on Mathematical Software (TOMS). 2008. Vol. 35, N. 2. P. 1—21.

13. **Hansen N.** et al. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Diss. INRIA, 2010.

14. **Decison** Tree for Optimization Software. URL: http://plato.asu.edu/guide.html (дата обращения: 25.01.2018).

15. **Momin J., Yang X. S.** A literature survey of benchmark functions for global optimization problems // Journal of Mathematical Modelling and Numerical Optimisation. 2013. Vol. 4, N. 2. P. 150—194.

16. **Gould N. I. M., Orban D., Toint P. L.** CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization // Computational Optimization and Applications. 2015. Vol. 60, N. 3. P. 545—557.

17. **Global** Optimization Test Problems. URL: http://www.mat.univie.ac.at/~neum/glopt/test.html (дата обращения: 25.01.2018).

18. **Dolan E. D., Moré J. J., Munson T. S.** Benchmarking optimization software with COPS 3.0 // Argonne National Lab, IL (US). 2004. No. ANL/MCS-TM-273.

19. **Floudas C. A.** et al. Handbook of test problems in local and global optimization // Springer Science & Business Media. 2013. Vol. 33.

20. **Beiranvand V., Hare W., Lucet Y.** Best practices for comparing optimization algorithms // Optimization and Engineering. 2017. Vol. 18, N. 4. P. 815—848.

21. **Dolan E. D., Moré J. J.** Benchmarking optimization software with performance profiles // Mathematical programming. Vol. 91, N. 2. P. 201—213.

22. **Gould N., Scott J.** A Note on Performance Profiles for Benchmarking Software // ACM Transactions on Mathematical Software (TOMS). 2016. Vol. 43, N. 2. P. 15.

23. **Moré J. J., Wild S. M.** Benchmarking derivative-free optimization algorithms // SIAM Journal on Optimization. 2009. Vol. 20, N. 1. P. 172—191.

24. **Bongartz I.** et al. CUTE: Constrained and unconstrained testing environment // ACM Transactions on Mathematical Software (TOMS). 1995. Vol. 21, N. 1. P. 123—160.

25. **Mersmann O.** et al. Exploratory landscape analysis // Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM. 2011. P. 829—836.

26. **Ugolotti R.** Meta-optimization of Bio-inspired Techniques for Object Recognition. Diss. Università di Parma. Dipartimento di Ingegneria dell'Informazione, 2015.

27. **Smit S. K., Eiben A. E., Szlávik Z.** An MOEA-based Method to Tune EA Parameters on Multiple Objective Functions // IJCCI (ICEC). 2010. P. 261—268.

28. **Li G., Aute V., Azarm S.** An accumulative error based adaptive design of experiments for offline metamodeling // Structural and Multidisciplinary Optimization. 2010. Vol. 40, N. 1. P. 137—155.

29. **Kerschke P.** et al. Cell mapping techniques for exploratory landscape analysis // EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V. Springer, Cham. 2014. P. 115—131.

30. **Flamm C.** et al. Barrier trees of degenerate landscapes // Zeitschrift für physikalische chemie. 2002. Vol. 216, N. 2. P. 155.

31. **Muñoz M. A., Kirley M., Halgamuge S. K.** Exploratory landscape analysis of continuous space optimization problems using information content // IEEE Transactions on Evolutionary Computation. 2015. Vol. 19, N. 1. Pp. 74—87.

32. **Preuss M.** Improved Topological Niching for Real-Valued Global Optimization // European Conference on the Applications of Evolutionary Computation. Springer, Berlin, Heidelberg. 2012. P. 386—395.

33. **Kerschke P.** et al. Detecting funnel structures by means of exploratory landscape analysis // Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. ACM. 2015. P. 265—272.

34. **Munoz M. A., Smith-Miles K.** Effects of function translation and dimensionality reduction on landscape analysis // Evolutionary Computation (CEC), 2015 IEEE Congress on. IEEE. 2015. P. 1336—1342.

35. **Eiben A. E., Michalewicz Z., Schoenauer M., Smith J. E.** Parameter Control in Evolutionary Algorithms // Parameter setting in evolutionary algorithms. Springer Berlin Heidelberg. 2007. P. 19—46.

36. **Pedersen M. E. H., Chipperfield A. J.** Parameter tuning versus adaptation: proof of principle study on differential evolution // HL0803. Hvass Laboratories. 2008.

37. **Smit S. K.** Parameter tuning and scientific testing in evolutionary algorithms. Vrije Universiteit, 2012.

38. **Mercer R. E., Sampson J. R.** Adaptive search using a reproductive meta-plan // Kybernetes. 1978. Vol. 7, N. 3. P. 215—228.

39. **Grefenstette J. J.** Optimization of control parameters for genetic algorithms // IEEE Transactions on systems, man, and cybernetics. 1986. Vol. 16, N. 1. P. 122—128.

40. **Hutter F., Hoos H. H., Stützle T.** Automatic algorithm configuration based on local search // Aaai. 2007. Vol. 7. P. 1152—1157.

41. **Yuan B., Gallagher M.** Combining Meta-EAs and racing for difficult EA parameter tuning tasks // Parameter Setting in Evolutionary Algorithms. Springer, Berlin, Heidelberg. 2007. P. 121—142.

42. **Hutter F., Hoos H. H., Leyton-Brown K.** Sequential model-based optimization for general algorithm configuration // International Conference on Learning and Intelligent Optimization. Springer, Berlin, Heidelberg. 2011. P. 507—523.

43. **Biau G.** Analysis of a random forests model // Journal of Machine Learning Research. 2012. Vol. 13, N. 1. P. 1063—1095.

44. **Myers R., Hancock E. R.** Empirical modelling of genetic algorithms // Evolutionary computation. 2001. Vol. 9, N. 4. P. 461—493.

45. **Taguchi G.** Taguchi methods: design of experiments. Vol. 4. Amer Supplier Inst, 1993.

46. **Adenso-Diaz B., Laguna M.** Fine-tuning of algorithms using fractional experimental designs and local search // Operations Research. 2006. Vol. 54, N. 1. P. 99—114.

47. **Pedersen M. E. H.** Tuning & simplifying heuristical optimization. Diss. University of Southampton, 2010.

48. **Goldsman D., Nelson B. L., Schmeiser B.** Methods for selecting the best system // Proceedings of the 23rd conference on Winter simulation. IEEE Computer Society. 1991. P. 177—186.

49. **Branke J., Chick S. E., Schmidt C.** New developments in ranking and selection: an empirical comparison of the three main approaches // Proceedings of the 37th conference on Winter simulation. Winter Simulation Conference. IEEE. 2005. P. 708—717.

50. **Schmeiser B.** Simulation experiments. Vol. 2 // Handbooks in operations research and management science. 1990. P. 295—330.

51. **Rinott Y.** On two-stage selection procedures and related probability-inequalities // Communications in Statistics-Theory and methods. 1978. Vol. 7, N. 8. P. 799—811.

52. **Hochberg Y., Tamhane A. C.** Multiple comparison procedures. New York, NY: John Wiley & Sons, Inc, 1987.

53. **Kim S. H., Nelson B. L.** A fully sequential procedure for indifference-zone selection in simulation // ACM Transactions on Modeling and Computer Simulation (TOMACS). 2001. Vol. 11, N. 3. P. 251—273.

54. **Balaprakash P., Birattari M., Stützle T.** Improvement strategies for the F-Race algorithm: Sampling design and iterative refinement // International workshop on hybrid metaheuristics. Springer, Berlin, Heidelberg. 2007. P. 108—122.

55. **Maron O., Moore A. W.** The racing algorithm: Model selection for lazy learners // Lazy learning. Springer, Dordrecht. 1997. P. 193—225.

56. **Birattari M., Kacprzyk J.** Tuning metaheuristics: a machine learning perspective. Berlin: Springer, 2009. Vol. 197. P. 85—114.

57. **Barros P. A. Jr., Kirby M. R., Mavris D. N.** Impact of sampling techniques selection on the creation of response surface models // SAE transactions. 2004. Vol. 113, N. 1. P. 1682—1693.

58. **Czarn A.** et al. Statistical exploratory analysis of genetic algorithms // IEEE Transactions on Evolutionary Computation. 2004. Vol. 8, N. 4. P. 405—421.

59. **Ramos I. C. O.** et al. Logistic regression for parameter tuning on an evolutionary algorithm // IEEE Congress on Evolutionary Computation. IEEE. 2005. Vol. 2. P. 1061—1068.

60. **Coy S. P.** et al. Using experimental design to find effective parameter settings for heuristics // Journal of Heuristics, Vol. 7, N. 1. P. 77—97.

61. **Bartz-Beielstein T., Parsopoulos K. E., Vrahatis M. N.** Analysis of particle swarm optimization using computational statistics // International conference on numerical analysis and applied mathematics ICNAAM. 2004. P. 34—37.

62. **Banzhaf W., Lasarczyk C.** Genetic programming of an algorithmic chemistry // In: Genetic Programming Theory and Practice II. Springer US, 2005. P. 175—190.

63. **Hutter F.** et al. Sequential model-based parameter optimization: An experimental investigation of automated and interactive approaches // Experimental Methods for the Analysis of Optimization Algorithms. Springer, Berlin, Heidelberg. 2010. P. 363—414.

64. **Ruxton G. D.** The unequal variance t-test is an underused alternative to Student's t-test and the Mann—Whitney U test // Behavioral Ecology. 2006. Vol. 17, N. 4. P. 688—690.

65. **Ugolotti R.** et al. GPU-based Automatic Configuration of Differential Evolution: a case study // Portuguese Conference on Artificial Intelligence. Springer, Berlin, Heidelberg. 2013. P. 114—125.

66. **Hutter F.** et al. ParamILS: an automatic algorithm configuration framework // Journal of Artificial Intelligence Research. 2009. Vol. 36, N. 1. P. 267—306.

67. **Nannen V., Eiben A. E.** A method for parameter calibration and relevance estimation in evolutionary algorithms // Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM. 2006. P. 183—190.

68. **Smit S. K., Eiben A. E.** Multi-problem parameter tuning using bonesa // Artificial Evolution. 2011. P. 222—233.

69. **Branke J., Schmidt C., Schmec H.** Efficient fitness estimation in noisy environments // Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc. 2001. P. 243—250.

70. **Zitzler E.** et al. SPEA2: Improving the strength Pareto evolutionary algorithm // TIK-report. 2001. Vol. 103. P. 95—100.

71. **Deb K.** et al. A fast and elitist multiobjective genetic algorithm: NSGA-II // IEEE transactions on evolutionary computation. 2002. Vol. 6, N. 2. P. 182—197.

72. **Deb K., Srinivasan A.** Innovization: Innovating design principles through optimization // Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM. Seattle, Washington, USA. 2006. P. 1629—1636.

73. **Branke J., Elomari J. A.** Meta-optimization for parameter tuning with a flexible computing budget // Proceedings of the 14th annual conference on Genetic and evolutionary computation. ACM. 2012. P. 1245—1252.

74. **Fitzgerald T.** et al. ReACT: Real-Time Algorithm Configuration through Tournaments // Seventh Annual Symposium on Combinatorial Search. 2014.

75. **Powell M. J. D.** The BOBYQA algorithm for bound constrained optimization without derivatives // Cambridge NA Report NA2009/06, University of Cambridge, Cambridge. 2009. P. 26—46.

76. **Adams B. M.** et al. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 user's manual, Sandia National Laboratories, Tech. Rep. SAND2010-2183, 2009.

77. **Csendes T., Pál L., Sendín J. O. H., Banga J. R.** The GLOBAL optimization method revisited // Optimization Letters. 2008. Vol. 2, N. 4. P. 445—454.

78. **Plantenga T. D.** Hopspack 2.0 user manual, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, SAND2009-6265, 2009.

79. **Powell M. J. D.** Developments of NEWUOA for minimization without derivatives // IMA journal of numerical analysis. 2008. Vol. 28, N. 4. P. 649—664.

80. **Le Digabel S., Tribes C.** NOMAD User Guide Version 3.5. Groupe d'études et de recherche en analyse des décisions. 2009. P. 1—45.

81. **Custódio A. L., Vicente L. N.** SID-PSM: A Pattern Search Method Guided by Simplex Derivatives for Use in Derivative-Free Optimization // Departamento de Matemática, Universidade de Coimbra, Coimbra. 2008.

82. **Optimization** Algorithms [Электронный ресурс] // modeFRONTIER by Esteco. URL: http://www.esteco.com/modefrontier/optimization-algorithms (дата обращения: 25.01.2018).

83. **Chase N.** et al. A benchmark study of optimization search algorithms // Red Cedar Technology. 2010.

84. **Generic** Tool for Optimization, GT Opt // Datadvance. URL: https://www.datadvance.net/product/pseven-core/generic-tool-for-optimization/ (дата обращения: 25.01.2018).

85. **Egorov I. N.** et al. IOSO optimization toolkit-novel software to create better design // 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. 2002. P. 4—6.

86. **Hutter F., Ramage S.** Manual for SMAC version v2. 10.02-master. Vancouver: Department of Computer Science University of British Columbia, 2015.

87. **Blot A.** et al. MO-ParamILS: A Multi-objective Automatic Algorithm Configuration Framework // International Conference on Learning and Intelligent Optimization. Springer International Publishing. 2016. P. 32—47.

88. **Ansótegui C., Sellmann M., Tierney K.** A gender-based genetic algorithm for the automatic configuration of algorithms // International Conference on Principles and Practice of Constraint Programming. Springer, Berlin, Heidelberg. 2009. P. 142—157.

89. **López-Ibánez M.** et al. The irace package: Iterated racing for automatic algorithm configuration // IRIDIA, Université Libre de Bruxelles, Belgium, Tech. Rep. TR/IRIDIA/2011-004, 2011.

90. **López-Ibánez M., Stutzle T.** The automatic design of multiobjective ant colony optimization algorithms // IEEE Transactions on Evolutionary Computation. 2012. Vol. 16, N. 6. P. 861—875.

91. **Dubois-Lacoste J., López-Ibáñez M., Stützle T.** Automatic configuration of state-of-the-art multi-objective optimizers using the TP + PLS framework // Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM. 2011. P. 2019—2026.

92. **Battiti R., Passerini A.** Brain-computer evolutionary multiobjective optimization (BC-EMO): a genetic algorithm adapting to the decision maker // IEEE Transaction on Evolutionary Computation. 2010. Vol. 14, N. 5. P. 671—687.

93. **Карпенко А. П., Моор Д. А., Мухлисуллина Д. Т.** Многокритериальная оптимизация на основе нейронечеткой аппроксимации функции предпочтений лица, принимающего решения // Наука и образование. 2010. № 6.

**T. A. Agasiev,** Ph.D. student, Assistant, email: agtaleh@mail.ru,
**A. P. Karpenko,** Doctor of Sc., Professor, email: apkarpenko@mail.ru
Bauman Moscow State Technical University, Moscow, 105005, Russian Federation

# Modern Techniques of Global Optimization. Review

*A global constrained optimization problem with high computational complexity of an objective function was formulated. The following terms have been defined: optimization problem features, base problem and base algorithm, meta-problem and meta-algorithm, base problem strategy, indicator of the strategy efficiency. Each optimization problem can be represented by a vector of its features, describing those properties of the problem and objective function that are meaningful in terms of optimization algorithms efficiency. The vector of problems features, estimated by means of exploratory landscape analysis methods, can be used to classify optimization problems or to gain important information about the problem definition in order to find an appropriate strategy of optimization algorithm to be used. To compare candidate strategies and select arguably the most suitable one the efficiency indicator should be determined. The article considers various ways of the estimation of optimization algorithms efficiency. The problem of finding the best algorithm's strategies based on a vector of problem's features and a given efficiency indicator is called meta-optimization problem. Statements of multi-indicator, multi-class and multi-budget meta-optimization problems are provided. The result of solving the meta-optimization problem is a set of the most appropriate strategies, i.e. the optimization algorithm tuned for solving particular types of optimization problems. Different approaches to automated tuning of optimization algorithms are described. A review of advanced exploratory landscape analysis and meta-optimization methods was performed based on approximately 100 articles. The article concludes with considering a modern software for solving the base and meta-optimization problems.*

***Keywords:*** *global optimization, exploratory landscape analysis, meta-optimization, surrogate modelling, performance analysis of optimization algorithms*

**References**

1. **Shan S., Wang G. G.** Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions, *Structural and Multidisciplinary Optimization*, 2010, vol. 41, no. 2, pp. 219—241.

2. **Wolpert D. H., Macready W. G.** No free lunch theorems for optimization, *IEEE transactions on evolutionary computation*, 1997, vol. 1, no. 1, pp. 67—82.

3. **Rios L. M., Sahinidis N. V.** Derivative-free optimization: a review of algorithms and comparison of software implementations, *Journal of Global Optimization*, 2013, vol. 56, no. 3, pp. 1247—1293.

4. **Kirkpatrick S.** Optimization by simulated annealing: Quantitative studies, *Journal of Statistical Physics*, 1984, vol. 34, no. 5—6, pp. 975—986.

5. **Liepins G. E., Hilliard M. R.** Genetic algorithms: Foundations and applications, *Annals of Operations Research*, 1989, vol. 21, no. 1, pp. 31—58.

6. **Hansen N.** The CMA evolution strategy: a comparing review, *Towards a new evolutionary computation*, Springer Berlin Heidelberg, 2006, pp. 75—102.

7. **Poli R., Kennedy J., Blackwell T.** Particle swarm optimization, *Swarm intelligence*, 2007, vol. 1, no. 1, pp. 33—57.

8. **Karaboga D., Basturk B.** A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 2007, vol. 39, no. 3, pp. 459—471.

9. **Dorigo M., Birattari M., Stutzle T.** Ant colony optimization, *IEEE Computational Intelligence Magazine*, 2006, vol. 1, no. 4, pp. 28—39.

10. **Jones D. R., Schonlau M., Welch W. J.** Efficient global optimization of expensive black-box functions, *Journal of Global Optimization*, 1998, vol. 13, no. 4, pp. 455—492.

11. **Cox D. D., John S.** SDO: A Statistical Method for Global Optimization, *Multidisciplinary Design Optimization: State of the Art*, 1997, vol. 80, pp. 315—329.

12. **Huyer W., Neumaier A.** SNOBFIT — Stable noisy optimization by branch and fit, *ACM Transactions on Mathematical Software (TOMS)*, 2008, vol. 35, no. 2, pp. 1—21.

13. **Hansen N.** et al. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Diss. INRIA, 2010.

14. **Decison** Tree for Optimization Software, available at: http://plato.asu.edu/guide.html (accessed 25.01.2018).

15. **Momin J., Yang X. S.** A literature survey of benchmark functions for global optimization problems, *Journal of Mathematical Modelling and Numerical Optimisation,* 2013, vol. 4, no. 2, pp. 150—194.

16. **Gould N. I. M., Orban D., Toint P. L.** CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization, *Computational Optimization and Applications*, 2015, vol. 60, no. 3, pp. 545—557.

17. **Global** Optimization Test Problems, available at: http:// www.mat.univie.ac.at/~neum/glopt/test.html (accessed 25.01.2018).

18. **Dolan E. D., Moré J. J., Munson T. S.** Benchmarking optimization software with COPS 3.0, *Argonne National Lab*, IL (US), 2004, no. ANL/MCS-TM-273.

19. **Floudas C. A.** et al. Handbook of test problems in local and global optimization, Springer Science & Business Media, 2013, vol. 33.

20. **Beiranvand V., Hare W., Lucet Y.** Best practices for comparing optimization algorithms, *Optimization and Engineering*, 2017, vol. 18, no. 4, pp. 815—848.

21. **Dolan E. D., Moré J. J.** Benchmarking optimization software with performance profiles, *Mathematical Programming*, vol. 91, no. 2, pp. 201—213.

22. **Gould N., Scott J.** A Note on Performance Profiles for Benchmarking Software, *ACM Transactions on Mathematical Software (TOMS),* 2016, vol. 43, no. 2, p. 15.

23. **Moré J. J., Wild S. M.** Benchmarking derivative-free optimization algorithms, *SIAM Journal on Optimization*, 2009, vol. 20, no. 1, pp. 172—191.

24. **Bongartz I.** et al. CUTE: Constrained and unconstrained testing environment, *ACM Transactions on Mathematical Software (TOMS),* 1995, vol. 21, no. 1, pp. 123—160.

25. **Mersmann O.** et al. Exploratory landscape analysis, *Proceedings of the 13th annual conference on Genetic and evolutionary computation. ACM*, 2011, pp. 829—836.

26. **Ugolotti R.** Meta-optimization of Bio-inspired Techniques for Object Recognition. Diss. Università di Parma. Dipartimento di Ingegneria dell'Informazione, 2015.

27. **Smit S. K., Eiben A. E., Szlávik Z.** An MOEA-based Method to Tune EA Parameters on Multiple Objective Functions, *IJCCI (ICEC)*, 2010, pp. 261—268.

28. **Li G., Aute V., Azarm S.** An accumulative error based adaptive design of experiments for offline metamodeling, *Structural and Multidisciplinary Optimization*, 2010, vol. 40, no. 1, pp. 137—155.

29. **Kerschke P.** et al. Cell mapping techniques for exploratory landscape analysis, *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation* V. Springer, Cham., 2014, pp. 115—131.

30. **Flamm C.** et al. Barrier trees of degenerate landscapes, *Zeitschrift für Physikalische Chemie*, 2002, vol. 216, no. 2, p. 155.

31. **Muñoz M. A., Kirley M., Halgamuge S. K.** Exploratory landscape analysis of continuous space optimization problems using information content, *IEEE Transactions on Evolutionary Computation*, 2015, vol. 19, no. 1, pp. 74—87.

32. **Preuss M.** Improved Topological Niching for Real-Valued Global Optimization, *European Conference on the Applications of Evolutionary Computation,* Springer, Berlin, Heidelberg, 2012, pp. 386—395.

33. **Kerschke P.** et al. Detecting funnel structures by means of exploratory landscape analysis, *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, 2015, pp. 265—272.

34. **Munoz M. A., Smith-Miles K.** Effects of function translation and dimensionality reduction on landscape analysis, *Evolutionary Computation (CEC), 2015 IEEE Congress on. IEEE*, 2015, pp. 1336—1342.

35. **Eiben A. E., Michalewicz Z., Schoenauer M., Smith J. E.** Parameter Control in Evolutionary Algorithms, *Parameter Setting in Evolutionary Algorithms*. Springer Berlin Heidelberg, 2007, pp. 19—46.

36. **Pedersen M. E. H., Chipperfield A. J.** Parameter tuning versus adaptation: proof of principle study on differential evolution, HL0803. Hvass Laboratories, 2008.

37. **Smit S. K.** Parameter tuning and scientific testing in evolutionary algorithms, Vrije Universiteit, 2012.

38. **Mercer R. E., Sampson J. R.** Adaptive search using a reproductive meta-plan, *Kybernetes*, 1978, vol. 7, no. 3, pp. 215—228.

39. **Grefenstette J. J.** Optimization of control parameters for genetic algorithms, *IEEE Transactions on systems, man, and cybernetics*, 1986, v,ol. 16, no. 1, pp. 122—128.

40. **Hutter F., Hoos H. H., Stützle T.** Automatic algorithm configuration based on local search, *Aaai*, 2007, vol. 7, pp. 1152—1157.

41. **Yuan B., Gallagher M.** Combining Meta-EAs and racing for difficult EA parameter tuning tasks, *Parameter Setting in Evolutionary Algorithms*, Springer, Berlin, Heidelberg, 2007, pp. 121—142.

42. **Hutter F., Hoos H. H., Leyton-Brown K.** Sequential model-based optimization for general algorithm configuration, *International Conference on Learning and Intelligent Optimization*, Springer, Berlin, Heidelberg, 2011, pp. 507—523.

43. **Biau G.** Analysis of a random forests model, *Journal of Machine Learning Research*., 2012, vol. 13, no. 1, pp. 1063—1095.

44. **Myers R., Hancock E. R.** Empirical modelling of genetic algorithms, *Evolutionary Computation*, 2001, vol. 9, no. 4, pp. 461—493.

45. **Taguchi G.** Taguchi methods: design of experiments, vol. 4. Amer Supplier Inst, 1993.

46. **Adenso-Diaz B., Laguna M.** Fine-tuning of algorithms using fractional experimental designs and local search, *Operations Research*, 2006, vol. 54, no. 1, pp. 99—114.

47. **Pedersen M. E. H.** Tuning & simplifying heuristical optimization, Diss. University of Southampton, 2010.

48. **Goldsman D., Nelson B. L., Schmeiser B.** Methods for selecting the best system, *Proceedings of the 23rd conference on Winter simulation. IEEE Computer Society*, 1991, pp. 177—186.

49. **Branke J., Chick S. E., Schmidt C.** New developments in ranking and selection: an empirical comparison of the three main approaches, *Proceedings of the 37th conference on Winter simulation. Winter Simulation Conference, IEEE,* 2005, pp. 708—717.

50. **Schmeiser B.** Simulation experiments. Vol. 2, *Handbooks in Operations Research and Management Science,* 1990, pp. 295—330.

51. **Rinott Y.** On two-stage selection procedures and related probability-inequalities, *Communications in Statistics-Theory and methods*, 1978, vol. 7, no. 8, pp. 799—811.

52. **Hochberg Y., Tamhane A. C.** Multiple comparison procedures. New York, NY, John Wiley & Sons, Inc, 1987.

53. **Kim S. H., Nelson B. L.** A fully sequential procedure for indifference-zone selection in simulation, *ACM Transactions on Modeling and Computer Simulation (TOMACS),* 2001, vol. 11, no. 3, pp. 251—273.

54. **Balaprakash P., Birattari M., Stützle T.** Improvement strategies for the F-Race algorithm: Sampling design and iterative refinement, *International workshop on hybrid metaheuristics*, Springer, Berlin, Heidelberg, 2007, pp. 108—122.

55. **Maron O., Moore A. W.** The racing algorithm: Model selection for lazy learners. Lazy learning, Springer, Dordrecht, 1997, pp. 193—225.

56. **Birattari M., Kacprzyk J.** Tuning metaheuristics: a machine learning perspective, Berlin, Springer, 2009, vol. 197, pp. 85—114.

57. **Barros P. A. Jr., Kirby M. R., Mavris D. N.** Impact of sampling techniques selection on the creation of response surface models, *SAE transactions*, 2004, vol. 113, no. 1, pp. 1682—1693.

58. **Czarn A.** et al. Statistical exploratory analysis of genetic algorithms, *IEEE Transactions on Evolutionary Computation*, 2004, vol. 8, no. 4, pp. 405—421.

59. **Ramos I. C. O.** et al. Logistic regression for parameter tuning on an evolutionary algorithm, *IEEE Congress on Evolutionary Computation. IEEE*, 2005, vol. 2, pp. 1061—1068.

60. **Coy S. P.** et al. Using experimental design to find effective parameter settings for heuristics, *Journal of Heuristics*, vol. 7, no. 1, pp. 77—97.

61. **Bartz-Beielstein T., Parsopoulos K. E., Vrahatis M. N.** Analysis of particle swarm optimization using computational statistics, *International conference on numerical analysis and applied mathematics ICNAAM*, 2004, pp. 34—37.

62. **Banzhaf W., Lasarczyk C.** Genetic programming of an algorithmic chemistry, *Genetic Programming Theory and Practice II*. Springer US, 2005, pp. 175—190.

63. **Hutter F.** et al. Sequential model-based parameter optimization: An experimental investigation of automated and interactive approaches, *Experimental Methods for the Analysis of Optimization Algorithms*, Springer, Berlin, Heidelberg, 2010, pp. 363—414.

64. **Ruxton G. D.** The unequal variance t-test is an underused alternative to Student's t-test and the Mann—Whitney U test, *Behavioral Ecology*, 2006, vol. 17, no. 4, pp. 688—690.

65. **Ugolotti R.** et al. GPU-based Automatic Configuration of Differential Evolution: a case study, *Portuguese Conference on Artificial Intelligence*, Springer, Berlin, Heidelberg, 2013, pp. 114—125.

66. **Hutter F.** et al. ParamILS: an automatic algorithm configuration framework, *Journal of Artificial Intelligence Research*, 2009, vol. 36, no. 1, pp. 267—306.

67. **Nannen V., Eiben A. E.** A method for parameter calibration and relevance estimation in evolutionary algorithms, *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. *ACM*, 2006, pp. 183—190.

68. **Smit S. K., Eiben A. E.** Multi-problem parameter tuning using bonesa, *Artificial Evolution*, 2011, pp. 222—233.

69. **Branke J., Schmidt C., Schmec H.** Efficient fitness estimation in noisy environments, *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, Morgan Kaufmann Publishers Inc., 2001, pp. 243—250.

70. **Zitzler E.** et al. SPEA2: Improving the strength Pareto evolutionary algorithm, *TIK-report*, 2001, vol. 103, pp. 95—100.

71. **Deb K.** et al. A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE transactions on evolutionary computation*, 2002, vol. 6, no. 2, pp. 182—197.

72. **Deb K., Srinivasan A.** Innovization: Innovating design principles through optimization, *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM. Seattle, Washington, USA, 2006, pp. 1629—1636.

73. **Branke J., Elomari J. A.** Meta-optimization for parameter tuning with a flexible computing budget, *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, ACM. 2012, pp. 1245—1252.

74. **Fitzgerald T.** et al. ReACT: Real-Time Algorithm Configuration through Tournaments, *Seventh Annual Symposium on Combinatorial Search*, 2014.

75. **Powell M. J. D.** The BOBYQA algorithm for bound constrained optimization without derivatives, *Cambridge NA Report NA2009/06*, University of Cambridge, Cambridge, 2009, pp. 26—46.

76. **Adams B. M.** et al. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 user's manual, Sandia National Laboratories, Tech. Rep. SAND2010-2183, 2009.

77. **Csendes T., Pál L., Sendín J. O. H., Banga J. R.** The GLOBAL optimization method revisited, *Optimization Letters*, 2008, vol. 2, no. 4, pp. 445—454.

78. **Plantenga T. D.** Hopspack 2.0 user manual, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, SAND2009-6265, 2009.

79. **Powell M. J. D.** Developments of NEWUOA for minimization without derivatives, *IMA Journal of Numerical Analysis,* 2008, vol. 28, no. 4, pp. 649—664.

80. **Le Digabel S., Tribes C.** NOMAD User Guide Version 3.5. Groupe d'études et de recherche en analyse des décisions, 2009, pp. 1—45.

81. **Custódio A. L., Vicente L. N.** SID-PSM: A Pattern Search Method Guided by Simplex Derivatives for Use in Derivative-Free Optimization, *Departamento de Matemática, Universidade de Coimbra*, Coimbra, 2008.

82. **Optimization** Algorithms (modeFRONTIER by Esteco), available at: http://www.esteco.com/modefrontier/optimization-algorithms (accessed 25.01.2018).

83. **Chase N.** et al. A benchmark study of optimization search algorithms, Red Cedar Technology, 2010.

84. **Generic** Tool for Optimization, GT Opt (Datadvance), available at: https://www.datadvance.net/product/pseven-core/generic-tool-for-optimization/ (accessed 25.01.2018).

85. **Egorov I. N.** et al. IOSO optimization toolkit-novel software to create better design, *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2002, pp. 4—6.

86. **Hutter F., Ramage S.** Manual for SMAC version v2. 10.02-master. Vancouver: Department of Computer Science University of British Columbia, 2015.

87. **Blot A.** et al. MO-ParamILS: A Multi-Objective Automatic Algorithm Configuration Framework, *International Conference on Learning and Intelligent Optimization*, Springer International Publishing, 2016, pp. 32—47.

88. **Ansótegui C., Sellmann M., Tierney K.** A gender-based genetic algorithm for the automatic configuration of algorithms, *International Conference on Principles and Practice of Constraint Programming*, Springer, Berlin, Heidelberg, 2009, pp. 142—157.

89. **López-Ibánez M.** et al. The irace package: Iterated racing for automatic algorithm configuration, IRIDIA, Université Libre de Bruxelles, Belgium, Tech. Rep. TR/IRIDIA/2011-004, 2011.

90. **López-Ibánez M., Stutzle T.** The automatic design of multiobjective ant colony optimization algorithms, *IEEE Transactions on Evolutionary Computation*, 2012, vol. 16, no. 6, pp. 861—875.

91. **Dubois-Lacoste J., López-Ibáñez M., Stützle T.** Automatic configuration of state-of-the-art multi-objective optimizers using the TP + PLS framework, *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011, pp. 2019—2026.

92. **Battiti R., Passerini A.** Brain-computer evolutionary multiobjective optimization (BC-EMO): a genetic algorithm adapting to the decision maker, *IEEE Transaction on Evolutionary Computation*, 2010, vol. 14, no. 5, pp. 671—687.

93. **Karpenko A. P., Moor D. A., Mukhlisullina D. T.** *Mnogokriterialnaja optimizacija na osnove nejro-nechetkoj approksimacii funkcii predpochtenij lica, prinimajushhego reshenija* (Multicriteria optimization based on neuro-fuzzy approximation of decision maker's utility function). *Science and Education of Bauman MSTU*, 2010, no. 6 (in Russian).