

А.П. Карпенко

СОВРЕМЕННЫЕ АЛГОРИТМЫ ПОИСКОВОЙ ОПТИМИЗАЦИИ

Алгоритмы, вдохновленные природой

*Допущено Учебно-методическим объединением вузов
по университетскому политехническому образованию
в качестве учебного пособия для студентов высших учебных
заведений, обучающихся по направлению подготовки
230100 «Информатика и вычислительная техника»*

 **ИЗДАТЕЛЬСТВО**
МГТУ им. Н.Э. БАУМАНА

Москва
2014

УДК 519.6
ББК 22.19
К26

Р е ц е н з е н т ы :

заведующий кафедрой «Информационные технологии в управлении»
Российской академии народного хозяйства и государственной службы
при Президенте Российской Федерации д-р техн. наук, проф. *А.Н. Данчул*;
профессор кафедры физико-математических методов управления
МГУ им. М.В. Ломоносова, Главный научный сотрудник лаборатории
оптимизации управляемых систем ИПУ им. В.А. Трапезникова РАН,
д-р техн. наук *Н.Б. Филимонов*

Карпенко А. П.

К26

Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой : учебное пособие / А. П. Карпенко. — Москва : Издательство МГТУ им. Н. Э. Баумана, 2014. — 446, [2] с. : ил.

ISBN 978-5-7038-3949-2

Учебное пособие посвящено, преимущественно, рассмотрению современных стохастических популяционных алгоритмов решения однокритериальной задачи оптимизации. Рассмотрены методы повышения эффективности этих алгоритмов путем их гибридизации и метаоптимизации. Наряду с однокритериальной рассматривается задача многокритериальной оптимизации и популяционные алгоритмы ее решения. Представлены методы распараллеливания указанных алгоритмов. Содержит большое число примеров решения тестовых и практически значимых задач оптимизации.

Для студентов высших учебных заведений, обучающихся по направлению 230100 «Информатика и вычислительная техника». Может быть полезно для всех студентов, изучающих курс «Методы оптимизации» и близкие по тематике курсы. Материал пособия представляет интерес также для аспирантов и специалистов, использующих в своей работе методы, алгоритмы и программы оптимизации.

УДК 519.6
ББК 22.19

ISBN 978-5-7038-3949-2

© Карпенко А. П., 2014
© Оформление. Издательство МГТУ
им. Н.Э. Баумана, 2014

*Моим любимым дочерям.
Главное, что вы есть!*

Предисловие

В последние годы интенсивно развиваются алгоритмы поисковой оптимизации, которые называют поведенческими, интеллектуальными, мета-эвристическими, вдохновленными (инспирированными) природой, роевыми, многоагентными, популяционными и т. д. Эффективность таких алгоритмов соизмерима, а часто превосходит эффективность ставших уже классическими эволюционных алгоритмов, среди которых наиболее известен генетический алгоритм. С помощью популяционных алгоритмов успешно решаются сложные оптимизационные задачи, например, задачи автоматизированного проектирования, синтеза сложных химических соединений, оптимального управления динамическими системами.

Большое число, прежде всего, англоязычных публикаций посвящено разработке, исследованию эффективности и практическому применению популяционных алгоритмов. В то же время для многих алгоритмов полностью или практически отсутствуют даже журнальные русскоязычные публикации. Данное учебное пособие призвано восполнить этот пробел и предоставить учащемуся широкий обзор современных популяционных алгоритмов поисковой оптимизации.

Основное внимание в пособии уделено популяционным алгоритмам глобальной оптимизации. Наряду с этим достаточно широко представлены алгоритмы этого класса, ориентированные на решение задачи многокритериальной оптимизации, точнее говоря – на приближенное построение множества и фронта Парето этой задачи. Поскольку практически значимые задачи оптимизации имеют, как правило, высокую вычислительную сложность, в пособии рассмотрены также вопросы реализации популяционных алгоритмов на параллельных вычислительных системах различной архитектуры. В целом, в пособии рассмотрены все основные аспекты популяционных алгоритмов – представлены канонические алгоритмы, большое число их модификаций, методы повышения эффективности алгоритмов путем гибридизации и метаоптимизации, методы распараллеливания вычислений. В пособии также представлено большое число примеров решения тестовых и практически значимых задач однокритериальной и многокритериальной оптимизации.

Пособие ориентировано, в первую очередь, на студентов высших учебных заведений, обучающихся по направлению 230100 «Информатика и вычислительная техника». В то же время оно может быть полезным для всех студентов, изучающих курс «Методы оптимизации» и близкие по тематике курсы. Материал пособия представляет интерес также для аспирантов и специалистов, использующих в своей работе методы, алгоритмы и программы оптимизации.

Пособие в значительной степени использует материалы лекций по курсу «Методы оптимизации», который автор в течение ряда лет читает студентам старших курсов специальности «Системы автоматизированного проектирования» МГТУ им. Н. Э. Баумана.

Автор благодарит своих аспирантов и студентов Абрамова Д. О., Антух А. Э., Бензу Н. Н., Березина Д. В., Гришина А. А., Егорова А. И., Карипова Д. Р., Комарова С. В., Кудряшова А. М., Митину Е. В., Мишина С. Н., Мурсякаева М. А., Мухлисуллина Э. Т., Овешникова А. А., Плякина Д. А., Семенихина А. С., Синяговскую О. А., Сорокина А. С., Сундукова К. В., Хасанову Р. В., Чернобривченко К. А., Шурова Д. Л. за помощь в проведении вычислительных экспериментов, результаты которых представлены в книге. Автор отмечает особенно большую помощь в подготовке книги Селиверстова Е. Ю. и выражает ему отдельную благодарность.

Модель объекта управления (п. 2.13) заимствована из книги: Федоренко Р. П. Приближенное решение задач оптимального управления. – М.: Наука, 1978. – 488 с. Задача, рассмотренная в п. 3.1.6, предоставлена И. М. Губайдуллиным (Институт нефтехимии и катализа РАН, г. Уфа). Модель системы «двигатель–генератор» взята из книги Александрова А. Г. Оптимальные и адаптивные системы. – М.: Высшая школа, 1989. – 263 с. Тестовая задача оптимизации конструкции консольной балки, рассмотренная в п. 5.1.3, заимствована из работы Lee K.S., Geem Z.W. *A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice // Computational Methods Applied Mechanics Engineering, 2005, Vol. 194, pp. 3902–3933.*

Основные обозначения

$abs(\cdot)$ – символ абсолютного значения числа

$D, D_X \subset R^{|X|}$ – множества допустимых значений вектора варьируемых параметров X

$D_F \subset R^{|F|}$ – множество достижимости многоцелевой задачи оптимизации

$D_F^* \subset D_F$ – фронт Парето многоцелевой задачи оптимизации

$D_X^* \subset D_X$ – множество Парето многоцелевой задачи оптимизации

$E(X) = (e_1(X), e_2(X), \dots, e_{|E|}(X))$ – $(|E| \times 1)$ -ограничивающая вектор-функция, с помощью которой задаются ограничения типа равенств на вектор варьируемых параметров X

$E_v(a, b)$ – $(v \times 1)$ -вектор независимых вещественных случайных чисел, распределенных по экспоненциальному закону $\xi(x) = \frac{1}{2b} \exp\left(-\frac{|x-a|}{b}\right)$, $a, b > 0$, $x \in (-\infty, \infty)$

$F(X) = (f_1(X), f_2(X), \dots, f_{|F|}(X))$ – целевая вектор-функция (векторный критерий оптимальности) со значениями в пространстве $R^{|F|}$

$F(X^*) = F^*$ – искомый целевой вектор

$f(X)$ – скалярная целевая функция (критерий оптимальности) со значениями в пространстве R^1

$f(X^*) = f^*$ – искомое оптимальное значение целевой функции $f(X)$

\tilde{f}^* – приближенное значение оптимума целевой функции $f(X)$

$G(X) = (g_1(X), g_2(X), \dots, g_{|G|}(X))$ – $(|G| \times 1)$ -ограничивающая вектор-функция, с помощью которой задаются ограничения типа неравенств на вектор варьируемых параметров X

$H = (h_1, h_2, \dots, h_{|H|})$ – хромосома (закодированный вектор X), соответствующая текущему положению агента в пространстве поиска

$H_i^s = (h_{i,j}, j \in [1 : |H_i^s|])$ – ген, соответствующий компоненте x_i вектора X

$L_v(\lambda)$ – $(v \times 1)$ -вектор независимых вещественных случайных чисел, распределенных по закону Леви; используется в полетах Леви

$N_v(m, \sigma)$ – $(v \times 1)$ -вектор независимых вещественных случайных чисел, распределенных по нормальному закону с математическим ожиданием m и средним квадратичным отклонением σ

n_f – число испытаний (вычислений значений целевой функции)

\bar{n}_f – среднее число испытаний

$Null$ – пустое множество

$R^{|A|}$ – $|A|$ -мерное арифметическое пространство

$S = S(t) = (s_i, i \in [1 : |S|])$ – текущая популяция

$s_i = s_i(t)$ – агент популяции S

$S' = S(t+1) = (s'_i, i \in [1 : |S|])$ – следующая популяция

$s'_i = s_i(t+1)$ – агент популяции S'

$\mathbf{S} = (S_i, i \in [1 : |\mathbf{S}|])$ – мультипопуляция

S_i – субпопуляция мультипопуляции \mathbf{S}

$\text{sign}(a)$ – знак числа a

t – символ транспонирования

\hat{t} – максимальное допустимое число поколений (итераций)

t – номер текущего поколения (итерации)

t_{end} – поколение (итерация), соответствующее началу стагнации вычислений

\bar{t}_{end} – среднее по числу запусков мултистарта число поколений (итераций)

$U_v(a; b)$ – $(v \times 1)$ -вектор независимых вещественных случайных чисел, равномерно распределенных в интервале $[a; b]$; a, b – вещественные числа; $b > a$

$U_v(n_1 : n_2)$ – $(v \times 1)$ -вектор независимых целых случайных чисел, равномерно распределенных в интервале $[n_1 : n_2]$; n_1, n_2 – целые числа; $n_2 > n_1$

u_{sign}^0 – случайная величина, с равной вероятностью принимающая значения $(-1), (0), (1)$

$u_{\text{sign}}^{\pm 1}$ – случайная величина, с равной вероятностью принимающая значения $(1), (-1)$

$X = (x_1, x_2, \dots, x_{|X|})$ – $(|X| \times 1)$ -вектор варьируемых параметров

$X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,|X|})$ – $(|X| \times 1)$ -вектор варьируемых параметров, определяющий положение агента $s_i \in S$; $i \in [1 : |X|]$

$X'_i = (x'_{i,1}, x'_{i,2}, \dots, x'_{i,|X|})$ – $(|X| \times 1)$ -вектор варьируемых параметров, определяющий положение агента $s'_i \in S'$; $i \in [1 : |X|]$

X^* – искомый оптимальный вектор варьируемых параметров X

\tilde{X}^* – приближенный вектор X^*

δ_t – критерий стагнации вычислительного процесса: если в течение данного числа поколений значение фитнес-функции не меняется, то считают, что имеет место стагнация вычислений

$\varphi(X) = \varphi(s) = \varphi(H)$ – фитнес-функция (функция приспособленности, приспособленность) агента s . Если не оговорено противное, то имеется в виду минимизация этой функции

$\chi(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0 \end{cases}$ – функция Хевисайда

$\{c_1, c_2, \dots\}$ – совокупность (множество) элементов c_1, c_2, \dots

$|A|$ – размерность вектора A

$|B|$ – число элементов во множестве B

$\lceil v \rceil$ – ближайшее целое большее v

$\lfloor v \rfloor$ – ближайшее целое меньшее v

$\lfloor v, w \rfloor$ – ближайшее целое меньшее v и кратное w

$\|\cdot\|$ – векторная норма

$\|\cdot\|_E$ – евклидова векторная норма

$\|A_1, A_2\|_E = \sqrt{\sum_{i=1}^{|A|} (a_{1,i} - a_{2,i})^2}$ – евклидово расстояние между целочисленными или вещественными векторами A_1, A_2 размерности $|A|$

$\|B_1, B_2\|_H$ – расстояния Хемминга между бинарными векторами B_1, B_2 , равное числу различающихся битов в одноименных разрядах указанных векторов

$\|A_1, A_2\|_M^1 = \sum_{i=1}^{|A|} abs(a_{1,i} - a_{2,i})$ – манхэттенское расстояние между целочисленными или вещественными векторами A_1, A_2 размерности $|A|$

$\|A_1, A_2\|_M^{|A|} = \left(\sum_{i=1}^{|A|} abs(a_{1,i} - a_{2,i})^{|A|} \right)^{\frac{1}{|A|}} = L_{|A|}$ -манхэттенское расстояние между целочисленными или вещественными векторами A_1, A_2 размерности $|A|$

$\nabla f(X') = \left(\left. \frac{\partial f(X)}{\partial x_1} \right|_{X=X'}, \left. \frac{\partial f(X)}{\partial x_2} \right|_{X=X'}, \dots, \left. \frac{\partial f(X)}{\partial x_{|X|}} \right|_{X=X'} \right)^T$ – градиент функции $f(X)$ в точке X'

(A, B) – скалярное произведение векторов A, B , имеющих одинаковую размерность

$A \otimes B = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \dots & a_1 b_{|B|} \\ a_2 b_1 & a_2 b_2 & \dots & a_2 b_{|B|} \\ \dots & \dots & \dots & \dots \\ a_{|A|} b_1 & a_{|A|} b_2 & \dots & a_{|A|} b_{|B|} \end{pmatrix}$ – прямое произведение векторов A, B одинаковой размерности $|A| = |B|$

$\Pi = \{X \mid X^- \leq X \leq X^+\} = \{X \mid x_i^- \leq x_i \leq x_i^+, i \in [1:|X|]\}$ – гиперпараллелепипед допустимых значений вектора варьируемых параметров

ЛПР – лицо, принимающее решения

ОДУ – система обыкновенных дифференциальных уравнений

Замечание. В силу того, что любой программный генератор случайных чисел в действительности генерирует псевдослучайные числа, вместо слов «случайный вектор», «случайная величина» и так далее следовало бы писать «псевдослучайный вектор», «псевдослучайная величина» и т. д. В книге для компактности письма этот нюанс опущен.

Введение

Многие задачи, возникающие в таких фундаментальных науках, как физика, химия и молекулярная биология, а также во многих прикладных науках, сводятся к задачам непрерывной глобальной оптимизации. Особенности таких задач часто являются нелинейность, недифференцируемость, многоэкстремальность (мульти-modalность), овражность, отсутствие аналитического выражения (плохая формализованность) и высокая вычислительная сложность оптимизируемых функций, высокая размерность пространства поиска, сложная топология области допустимых значений и т. д.

С самой общей точки зрения, именно указанные особенности задач глобальной оптимизации объясняют отсутствие универсального алгоритма их решения и, напротив, наличие чрезвычайно большого числа алгоритмов, их модификаций и гибридизаций. Число таких алгоритмов увеличивают также их параллельные версии и новые параллельные алгоритмы оптимизации, ориентированные на различные классы параллельных вычислительных систем.

Для эффективного решения задач глобальной оптимизации в 1980-х гг. начали интенсивно разрабатывать класс стохастических поисковых алгоритмов оптимизации, которые в разных публикациях называют поведенческими, интеллектуальными, метаэвристическими, вдохновленными (инспирированными) природой, роевыми, многоагентными, популяционными и т. д. Последний термин наиболее отвечает сути этих алгоритмов.

Популяционные алгоритмы (*population algorithms*) предполагают одновременную обработку нескольких вариантов решения задачи оптимизации и представляют собой альтернативу классическим «траекторным» поисковым алгоритмам, в которых в области поиска эволюционирует только один кандидат на решение этой задачи.

Подавляющее большинство рассматриваемых алгоритмов опубликовано в англоязычной литературе, в которой вместо традиционного для русскоязычного читателя термина «метод» принято использовать термин «алгоритм». Для того чтобы избежать возможной неоднозначности идентификации рассматриваемых объектов, также используется последний термин, хотя он с точки зрения русскоязычных публикаций не совсем корректен.

Задачи глобальной оптимизации делятся на два класса – детерминированные задачи и стохастические задачи. В первом случае оптимизируемая функция и функции, ограничивающие область решения задачи, являются детерминированными, то есть не содержат случайных параметров. Во втором случае одна или несколько указанных функций содержат такие параметры. В книге рассматриваются только детерминированные задачи оптимизации.

С другой точки зрения, среди задач глобальной оптимизации выделяют статические (стационарные) и динамические (нестационарные) задачи. В статических задачах оптимизируемая функция и область ее допустимых значений не меняются во времени, так что остаются неизменными положения в области

поиска ее локальных и глобальных экстремумов. В динамических задачах положения указанных экстремумов могут меняться во времени, и задача состоит в отслеживании движения глобального экстремума. Книга посвящена статическим задачам оптимизации, хотя многие из рассматриваемых популяционных алгоритмов могут быть модифицированы (и такие модификации известны) для решения также динамических задач оптимизации.

Таким образом, предметом рассмотрения являются детерминированные статические задачи глобальной безусловной (без ограничений на значения варьируемых параметров) и условной (с ограничениями на эти значения) непрерывной оптимизации. Последнее означает, что допустимыми значениями параметров являются вещественные числа.

По способу определения направления движения к экстремуму алгоритмы поисковой оптимизации разделяют на алгоритмы детерминированного (регулярного) поиска (например, классический алгоритм наискорейшего спуска) и алгоритмы стохастического (случайного) поиска. Все рассматриваемые популяционные алгоритмы относятся к классу стохастических.

Алгоритмы поисковой оптимизации разделяют еще на два следующих класса: алгоритмы, использующие как пробные, так и рабочие шаги поиска; алгоритмы, в которых пробные и рабочие шаги совмещены. Представленные в книге популяционные алгоритмы реализуют оба эти варианта поиска. Заметим, что в алгоритмах решения динамических задач оптимизации используют только совмещение пробных и рабочих шагов.

Все популяционные алгоритмы относятся к классу эвристических алгоритмов (*heuristic algorithms*), то есть алгоритмов, для которых сходимость к глобальному решению не доказана, но экспериментально установлено, что в большинстве случаев они дают достаточно хорошее решение.

В качестве общего названия членов популяции используем термин агент (*agent*). В различных популяционных алгоритмах агенты называются индивидами, особями, частицами, муравьями, пчелами и т. д.

Общая схема популяционных алгоритмов включает в себя следующие этапы.

1. *Инициализация популяции.* В области поиска тем или иным образом создаем некоторое число начальных приближений к искомому решению задачи — инициализируем популяцию агентов.

2. *Миграция агентов популяции.* С помощью некоторого набора миграционных операторов, специфических для каждого из популяционных алгоритмов, перемещаем агентов в области поиска таким образом, чтобы в конечном счете приблизиться к искомому экстремуму оптимизируемой функции.

3. *Завершение поиска.* Проверяем выполнение условий окончания итераций, и если они выполнены, завершаем вычисления, принимая лучшее из найденных положений агентов популяции за приближенное решение задачи. Если указанные условия не выполнены, возвращаемся к выполнению этапа 2.

При инициализации популяции могут быть использованы детерминированные и случайные алгоритмы. Формирование начальной популяции, агенты которой находятся вблизи глобального экстремума оптимизируемой функции, может существенно сократить время решения задачи. Однако обычно априорная

информация о местоположении этого экстремума отсутствует, поэтому агентов начальной популяции распределяют равномерно по всей области поиска.

Миграцию агентов, например, в генетическом алгоритме, который мы также относим к популяционным алгоритмам, реализуют генетические операторы скрещивания, мутации и другие.

В качестве условия окончания поиска используют, как правило, условие достижения заданного числа итераций (*поколений*). Часто используют также условие стагнации (*stagnation*) алгоритма, когда лучшее достигнутое значение оптимизируемой функции не изменяется в течение заданного числа поколений. Могут быть использованы и другие условия, например, условие исчерпания времени, отпущенного на решение задачи.

Из представленной общей схемы популяционных алгоритмов следует, что они обладают ярко выраженной модульной структурой, позволяющей легко получить большое число вариантов любого из алгоритмов путем варьирования и комбинирования правил инициализации популяции, миграционных операторов и условий завершения поиска.

Агенты популяции обладают следующими основными свойствами.

- Автономность – агенты движутся в пространстве поиска, хотя бы частично, независимо друг от друга.
- Стохастичность – процесс миграции агентов содержит случайную компоненту.
- Ограниченность представления – каждый из агентов популяции обладает информацией лишь об исследуемой им части области поиска и, быть может, об окружении некоторых других агентов.
- Децентрализация – отсутствие агентов, управляющих процессом поиска в целом.
- Коммуникабельность – агенты тем или иным способом могут обмениваться между собой информацией о топологии (ландшафте) оптимизируемой функции, выявленной в процессе исследования своей части области поиска.

Даже если стратегия поведения каждого из агентов популяции достаточно проста, указанные свойства агентов обеспечивают формирование так называемого роевого интеллекта (*swarm intelligence*) популяции, проявляющегося в самоорганизации и сложном поведении популяции в целом.

Одной из особенностей популяционных алгоритмов оптимизации является то, что в подавляющем большинстве случаев для них имеется достаточно любопытная аналогия в человеческом обществе, живой или неживой природе. Так, известны популяционные алгоритмы эволюции разума, колонии муравьев, роя медоносных пчел, светлячков, гравитационного и электромагнитного поиска и т. д.

Популяционные алгоритмы поисковой оптимизации в сравнении с классическими алгоритмами имеют неоспоримые преимущества, прежде всего, при решении задач высокой размерности, мультимодальных и плохо формализованных задач. В этих условиях популяционные алгоритмы могут обеспечить высокую вероятность локализации глобального экстремума оптимизируемой функции. Важно также, что популяционные алгоритмы позволяют эффективнее классических алгоритмов отыскать субоптимальное (близкое к оптимальному) решение. Часто достаточным является именно такое решение.

Популяционные алгоритмы почти наверняка проиграют алгоритму полного перебора при решении задачи оптимизации невысокой размерности. В случае гладкой и унимодальной оптимизируемой функции популяционные алгоритмы, как правило, менее эффективны, чем любой классический градиентный метод. К недостаткам популяционных алгоритмов следует отнести также сильную зависимость их эффективности от значений свободных параметров, число которых в большинстве алгоритмов достаточно велико.

Важно понимать, что для решения любой задачи оптимизации во множестве всех известных оптимизационных алгоритмов, включая популяционные алгоритмы, наверняка найдется хотя бы один алгоритм, который даст, как минимум, удовлетворительные результаты. Однако нет и не может быть алгоритма, который мог бы обеспечить высокую эффективность при решении всех задач оптимизации. Поэтому для каждого из популяционных алгоритмов могут быть выделены классы задач, которые он решает

- лучше других алгоритмов,
- примерно как другие алгоритмы,
- хуже других алгоритмов.

Важнейшим понятием популяционных алгоритмов является понятие фитнес-функции (*fitness-function*). Часто эту функцию называют функцией пригодности, функцией полезности, функцией приспособленности и т. д. Важность функции обусловлена тем обстоятельством, что с ее помощью оценивают «качество» агентов популяции. Стратегически, в процессе миграции агенты движутся таким образом, чтобы приблизиться к глобальному экстремуму фитнес-функции.

Часто, но далеко не всегда, фитнес-функция совпадает с оптимизируемой функцией. В общем случае в качестве фитнес-функции используют некоторое детерминированное или стохастическое, линейное или нелинейное преобразование оптимизируемой функции. Простейший пример – применение в качестве фитнес-функции инвертированной оптимизируемой функции. Более сложный пример – использование для построения фитнес-функции известного метода штрафных функций.

С использованием понятия фитнес-функции можно сказать, что суть популяционных алгоритмов состоит в обеспечении более высокой средней приспособленности положений агентов данного поколения по сравнению с их приспособленностью в предыдущем поколении.

Одна из основных проблем конструирования популяционных алгоритмов – обеспечение баланса между интенсивностью поиска (скоростью сходимости алгоритма) и широтой поиска (диверсификацией поиска). Интенсификация (*intensification*) поиска требует быстрой сходимости алгоритма, что означает быстрое уменьшение разнообразия популяции. Напротив, диверсификация (*diversification*) поиска призвана обеспечить более широкий обзор пространства поиска и более высокую вероятность локализации глобального экстремума задачи. Диверсификация требует сохранения разнообразия популяции в течение как можно большего числа поколений популяционного алгоритма.

Первые поколения популяционного алгоритма преимущественно занимаются исследованием области поиска, чтобы найти новых кандидатов на решение

задачи. Агенты последних поколений алгоритма заняты исключительно уточнением найденных ранее решений или решения. Промежуточные поколения в некоторой мере совмещают разведку и эксплуатацию разведанных решений.

Наиболее развитыми механизмами решения проблемы баланса между интенсивностью и ширитой поиска являются механизмы адаптации (*adapting*) и самоадаптации (*self-adapting*) популяционных алгоритмов. Эти механизмы реализуют изменение значений свободных параметров алгоритмов так, чтобы обеспечить постепенный переход от диверсификации к интенсификации поиска.

Поскольку популяционные алгоритмы являются стохастическими, их эффективность, как правило, меняется в широких пределах в зависимости от удачности начального приближения, полученного на этапе инициализации популяции. В этой связи для оценки эффективности данных алгоритмов используют многократные прогоны алгоритма, исходя из разных начальных приближений (метод мултистарта). Основными критериями эффективности популяционных алгоритмов являются надежность алгоритма – оценка вероятности локализации глобального экстремума, а также скорость его сходимости – оценка математического ожидания необходимого числа испытаний (вычислений значения оптимизируемой функции).

Можно предложить несколько классификаций популяционных алгоритмов. На верхнем иерархическом уровне выделяют следующие классы таких алгоритмов:

- эволюционные алгоритмы, включая генетические алгоритмы;
- популяционные алгоритмы, вдохновленные живой природой;
- алгоритмы, вдохновленные неживой природой;
- алгоритмы, инспирированные человеческим обществом;
- прочие алгоритмы.

Первая глава книги содержит постановку детерминированной задачи непрерывной оптимизации и классификацию алгоритмов ее решения. Здесь же рассмотрены некоторые классические непопуляционные стохастические алгоритмы решения задачи локальной безусловной оптимизации, аналогичные алгоритмы решения задачи условной оптимизации, а также основные неклассические алгоритмы глобальной оптимизации.

Во второй главе рассмотрены генетический алгоритм и другие эволюционные алгоритмы.

Третья глава освещает наиболее известные и развитые популяционные алгоритмы, относящиеся к классу алгоритмов, вдохновленных живой природой.

В четвертой главе рассматриваются менее известные и развитые популяционные алгоритмы, вдохновленные живой природой.

Пятая глава посвящена популяционным алгоритмам, инспирированным неживой природой, человеческим обществом, и некоторым другим алгоритмам, которые трудно отнести к одному из указанных классов.

В настоящее время основным средством конструирования новых высокоэффективных популяционных алгоритмов является гибридизация (*hybridization*) популяционного алгоритма с двумя и более популяционными и/или не популяционными алгоритмами. В шестой главе рассматривается классификация ме-

тодов гибридизации, общие принципы гибридизации, приводятся примеры основных классов гибридных популяционных алгоритмов.

Как отмечалось выше, одной из особенностей популяционных алгоритмов является наличие в них значительного, а часто очень большого числа свободных параметров. От значений этих параметров может сильно зависеть эффективность алгоритма, однако формальные рекомендации по выбору значений этих параметров, исходя из особенностей решаемой задачи оптимизации, как правило, отсутствуют. В этой связи в настоящее время интенсивно развиваются методы адаптации и самоадаптации значений этих параметров, называемые методами мета-оптимизации (*meta-optimization*). Данная проблематика рассматривается в седьмой главе.

При решении практических, прежде всего, технических задач оптимизации часто возникает задача многоцелевой (*multi objective*) оптимизации, когда желательно экстремизировать не одну, а несколько целевых функций. Хорошо известно, что, поскольку нельзя добиться экстремального значения всех целевых функций одновременно, решение такой задачи представляет собой некоторый компромисс, принадлежащий так называемому множеству Парето (*Pareto set*) задачи, то есть множеству решений, не улучшаемых одновременно по всем оптимизируемым функциям. Часто решением задачи многоцелевой оптимизации называют множество Парето этой задачи, предполагая, что после построения некоторой аппроксимации данного множества лицо, принимающее решение, неформальными или формализованными методами выберет из него единственное решение. Популяционные алгоритмы решения (в указанном смысле) задачи многоцелевой оптимизации рассматриваются в восьмой главе. Глава содержит постановку задачи многоцелевой оптимизации, включая формальное определение множества Парето, классификацию алгоритмов решения этой задачи, а также обзор популяционных алгоритмов построения аппроксимации множества Парето.

В связи с высокой вычислительной сложностью практически значимых задач оптимизации и быстро возрастающей доступностью различных высоко- и сверхвысокопроизводительных вычислительных систем, все более актуальной становится проблема разработки параллельных популяционных алгоритмов, ориентированных на различные классы параллельных ЭВМ. Этим алгоритмам посвящена девятая глава, в которой рассмотрена структура основных классов параллельных ЭВМ, приведена классификация методов синтеза параллельных алгоритмов оптимизации, а также примеры таких алгоритмов.

Во всех случаях в книге изложены, по сути, не алгоритмы, а лишь их схемы. Детали алгоритмов, быть может, весьма существенные, но затеняющие их суть, опущены. Поэтому материал книги не следует использовать непосредственно для программной реализации представленных популяционных алгоритмов – следует обращаться к оригинальным публикациям авторов этих алгоритмов.

Сводка основных популяционных алгоритмов, рассмотренных в работе, приведена в Приложении А.

Основные обозначения, используемые в книге, стандартизованы и почти всюду не совпадают с исходными авторскими обозначениями.

Глава 1

ПОСТАНОВКА ЗАДАЧИ ПОИСКОВОЙ ОПТИМИЗАЦИИ И НЕПОПУЛЯЦИОННЫЕ СТОХАСТИЧЕСКИЕ АЛГОРИТМЫ ЕЕ РЕШЕНИЯ

В первом параграфе главы даны общая постановка детерминированной задачи оптимизации, классификация этих задач по различным классификационным признакам, а также классификация алгоритмов их решения. Во втором и третьем параграфах рассмотрены непопуляционные стохастические алгоритмы решения задач, соответственно, локальной безусловной оптимизации и локальной условной оптимизации. Четвертый параграф посвящен таким же алгоритмам решения задачи глобальной оптимизации.

1.1. ПОСТАНОВКА И КЛАССИФИКАЦИЯ АЛГОРИТМОВ РЕШЕНИЯ ДЕТЕРМИНИРОВАННОЙ ЗАДАЧИ ПОИСКОВОЙ ОПТИМИЗАЦИИ

1.1.1. Постановка задачи

Рассмотрим *детерминированную задачу оптимизации*

$$\min_{X \in D \subset R^{|X|}} f(X) = f(X^*) = f^*, \quad (1.1)$$

где $|X|$ – размерность вектора варьируемых параметров $X = (x_1, x_2, \dots, x_{|X|})^T$; $R^{|X|}$ – $|X|$ -мерное арифметическое пространство; $f(X)$ – целевая функция (критерий оптимальности) со значениями в пространстве R^1 ; X^* , f^* – искомые оптимальное решение и значение целевой функции соответственно;

$$D = \{X \mid E(X) = 0, G(X) \geq 0\} \quad - \quad (1.2)$$

множество допустимых значений вектора X . Здесь $E(X)$ – ограничивающая вектор-функция с компонентами $e_1(X), e_2(X), \dots, e_{|E|}(X)$, а $G(X)$ – аналогичная функция с компонентами $g_1(X), g_2(X), \dots, g_{|G|}(X)$.

Детерминированность поставленной задачи оптимизации означает, что целевая функция $f(X)$ и ограничивающие функции $E(X)$, $G(X)$ не содержат в себе случайных параметров.

Целевую функцию $f(x)$, где $x \in [a; b]$ – скаляр, называют *унимодальной*, если в области ее определения $[a; b]$ существует точка $x_* \in [a; b]$ такая, что на полуинтервале $[a; x_*)$ функция $f(x)$ убывает, а на полуинтервале $(x_*; b]$ – возрастает. Точка x_* может быть как внутренней точкой отрезка $[a; b]$, так и совпадать с одним из его концов. Определение унимодальности целевой функции не требует ее непрерывности.

$$X^t, X^{t-1}, \dots, X^{t-k}, \quad (t-k) \geq 0, \quad (1.33)$$

и запрещает часть окрестности d^t текущего решения X^t , так что новая точка X^{t+1} является решением вспомогательной задачи локальной условной оптимизации

$$f(X^{t+1}) = \min_X f(X), \quad X \in d^t \setminus T^t. \quad (1.34)$$

Очевидно, что алгоритм *TS* трансформируется в обычный алгоритм локального поиска в случае, когда список T^t пуст, то есть когда $k = 0$.

Элементы списка T^t называют *табу-перемещениями* (*tabu moves*). Вообще говоря, список запретов может иметь переменную длину, когда $k = k(t) = k^t$. Величина k^t имеет смысл числа итераций, в течение которых данное решение находится в списке T^t , и называется *табу-продолжительностью* (*tabu tenure*). Известна рекомендация, по которой $k^t = O(|X|)$.

Общая схема алгоритма *TS* имеет следующий вид.

1) Тем или иным образом выбираем начальную точку X^0 и полагаем $\tilde{X}^* = X^0$, $\tilde{\varphi}^* = \varphi(X^0)$, $T^0 = \text{Null}$, $t = 0$.

2) Если $d^t \setminus T^t = \text{Null}$, то переходим к шагу 4. В противном случае, по формуле (1.34) находим решение X^{t+1} и полагаем $t = t + 1$.

3) Если $\varphi(X^t) < \tilde{\varphi}^*$, то выполняем присваивания $\tilde{X}^* = X^t$, $\tilde{\varphi}^* = \varphi(X^t)$.

4) Если условия окончания поиска выполнены, завершаем вычисления. В противном случае обновляем список запретов T^t и возвращаемся к шагу 2.

Особенностью условий окончания поиска в алгоритме *TS* является то, что эти условия включают в себя пустоту множества $d^t \setminus T^t$.

Слишком строгие запреты могут привести к преждевременной стагнации вычислительного процесса. Поэтому в алгоритме *TS* эпизодически разрешается нарушать запреты. С этой целью используют механизм так называемых *многообещающих решений* (*aspirant decisions*) или эквивалентный механизм *функции аспирации* (*aspirant-function*) $a(X)$, которая отменяет табу-статус перемещения X . Говорят, что решение $X \in T^t$ достигло уровня аспирации и может быть объявлено допустимым, если $\varphi(X) \leq a(X)$.

Вопросы для самопроверки

1. Приведите постановку детерминированной задачи оптимизации. Определите основные классы целевых функций.

2. Дайте классификацию задач оптимизации по виду целевой и ограничивающих функций, наличию и отсутствию ограничений, а также по их характеру, размерности вектора варьируемых параметров, числу точек минимума, характеру искомого решения.

3. Дайте классификацию поисковых алгоритмов оптимизации по характеру искомого решения, характеру ограничений, виду итерационной функции, числу предыдущих учитываемых шагов, порядку используемых производных.

4. Приведите итерационные формулы основных одношаговых алгоритмов случайного поиска, а также правила выбора величины шага в этих алгоритмах.

5. Представьте общую итерационную формулу многошаговых алгоритмов случайного поиска.

6. Изобразите траекторию поиска минимума некоторой двумерной овражной функции с помощью алгоритма комплексов (конфигураций).

7. Изобразите траекторию поиска минимума двумерной квадратичной функции с помощью алгоритма гиперболы (алгоритма случайного поиска с постоянным радиусом поиска и случайными направлениями).

8. Изложите общую схему алгоритмов последовательной безусловной оптимизации. Приведите схему алгоритма штрафных функций.

9. Назовите отличительные особенности метода скользящего допуска. Объясните суть явления осцилляции решений в этом методе.

10. Перечислите основные алгоритмы условной оптимизации, не использующие редукцию этой задачи к задаче безусловной оптимизации.

11. Представьте схему модифицированного алгоритма комплексов для решения задачи условной оптимизации.

12. Изложите схемы алгоритмов одномерных вложений и развертки Пеано.

13. Перечислите основные алгоритмы чисто случайного поиска. Изложите суть алгоритмов Монте-Карло, имитации отжига, хаотического алгоритма локального поиска, а также алгоритма на основе полетов Леви.

14. Дайте общее определение двухфазных алгоритмов случайного поиска. Укажите основные особенности алгоритмов мультистарта, жадного адаптивного алгоритма случайного поиска, алгоритма итерационного локального поиска.

15. Изложите суть алгоритма поиска с запретами.

Глава 2

ЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ

Эволюционные алгоритмы включают в себя генетические алгоритмы (*genetic algorithm*), эволюционную стратегию (*evolution strategy*), эволюционное программирование (*evolutionary programming*), алгоритмы дифференциальной эволюции (*differential evolution*), а также генетическое программирование (*genetic programming*). Последний класс алгоритмов не используют для решения задач непрерывной оптимизации, поэтому он далее не рассматривается.

Суть парадигмы эволюционных алгоритмов состоит в использовании базовых принципов теории биологической эволюции – отбора, мутации и воспроизводства особей. Эволюционные алгоритмы являются частью более широкой технологии *мягких вычислений* (*soft computing*), включающих в себя еще *нечёткую логику* Заде (*L. A. Zadeh*), *нейронные сети*, *вероятностные рассуждения* и *сети доверия*. Данные технологии дополняют друг друга и используются в различных комбинациях или самостоятельно для создания интеллектуальных систем.

Наиболее развитый класс эволюционных алгоритмов – генетические алгоритмы. Также большое развитие получили эволюционная стратегия и эволюционное программирование. В сравнении с указанными алгоритмами, алгоритм дифференциальной эволюции разработан в существенно меньшей степени.

Алгоритмы эволюционной стратегии и эволюционного программирования имеют ряд важных общих свойств, так что существует даже точка зрения, в соответствии с которой оба эти алгоритма, в сущности, одинаковы и представляют собой модификации генетического алгоритма. При разработке алгоритмов эволюционной стратегии и эволюционного программирования в качестве базовой принята следующая концепция: эволюция популяции заключается прежде всего в приспособлении особей популяции к окружающей среде на поведенческом уровне, а не на генетическом уровне, как в генетическом алгоритме. Следствием этой концепции является отсутствие в указанных методах рекомбинации особей, а значит, и оператора кроссовера. Единственным оператором в этих методах, обеспечивающим поиск альтернативных решений, является оператор мутации.

Глава посвящена последовательным неадаптивным эволюционным алгоритмам для решения одноцелевой задачи оптимизации. В первом параграфе главы изложены биологические предпосылки и общая схема эволюционных алгоритмов. Второй параграф посвящен методам кодирования особей в эволюционных алгоритмах. В трех последующих параграфах рассмотрены основные операторы эволюционных алгоритмов – операторы мутации, скрещивания и отбора особей. Шестой параграф содержит описание некоторых других операторов и процедур, используемых в эволюционных алгоритмах.

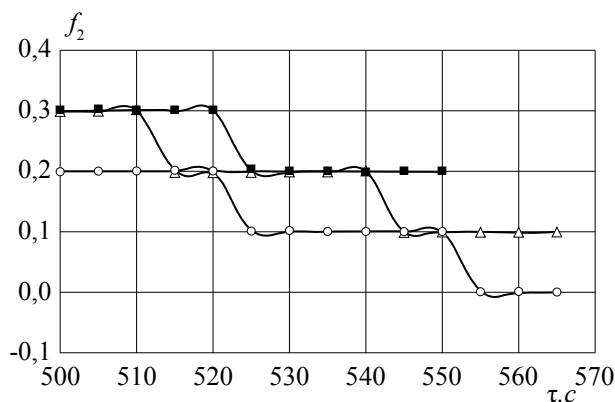


Рис. 2.16. Характер изменения минимального отклонения космического аппарата от заданной точки на поверхности Земли ($u^+ = 0,4$): ■ — одно поколение; \triangle — 30 поколений; \circ — 70 поколений

случая $u^+ = 0,4$ характер полученного решения иллюстрирует рис. 2.16, который показывает, что найденное оптимальное управление гарантирует точное попадание аппарата в заданную точку на поверхности Земли.

Вопросы для самопроверки

1. Представьте схему канонического генетического алгоритма.
2. Дайте определение монохромосомного, мультихромосомного, адаптивного, символического и вещественного кодирования хромосом.
3. Приведите примеры бинарных операторов мутации, использующих инверсию одного или нескольких битов родительской особи, а также операторов, основанных на перестановке битов этой особи.
4. Назовите отличительные признаки следующих вещественных операторов мутации: оператор случайной мутации, гауссов оператор мутации, арифметический оператор вещественного сдвига, геометрический оператор вещественного сдвига, *BGA*-оператор мутации, степенной оператор мутации, неравномерный мутатор Михалевича, динамический оператор мутации.
5. Перечислите основные двухродительские и многородительские бинарные кроссоверы.
6. Дайте определение следующих вещественных кроссоверов, которые на основе двух родительских особей продуцируют одну особь-потомка: плоский кроссовер, эвристический кроссовер, смешанный кроссовер, расширенный линейный кроссовер, дискретный кроссовер.
7. Назовите основные особенности следующих вещественных кроссоверов с двумя особями-потомками: простейший кроссовер, арифметический кроссовер, геометрический кроссовер, нечеткий кроссовер; кроссовер, имитирующий бинарный кроссовер.
8. Приведите пример вещественного кроссовера с тремя особями-потомками.
9. Определите следующие методы управления популяцией, учитывающие только приспособленность особей: метод рулетки, метод пропорционального отбора с остатком, метод стохастического универсального отбора, метод турнирного отбора, метод рангового отбора, метод отбора на основе элитизма.

10. Приведите пример метода управления популяцией, учитывающего как приспособленность особей, так и близость их генотипов.
11. Дайте определение функции отбора. Назовите основные классы функций отбора.
12. Приведите примеры следующих методов селекции: случайный метод селекции, не учитывающий приспособленность и генотип особей; метод, использующий только приспособленность особей; метод, основанный на генотипе особей.
13. Какие методы масштабирования фитнес-функции вы знаете?
14. Дайте определения следующих методов нишевания: метод разделяющей (сепарирующей) функции, метод детерминированного сжатия, метод последовательного нишевания.
15. Представьте схемы следующих типовых генетических алгоритмов: канонический генетический алгоритм, генетический алгоритм *Genitor*, генетический алгоритм *CHC*, генетический микроалгоритм.
16. Приведите определения основных понятий теории шим. Сформулируйте основную и модифицированную теоремы шим.
17. Назовите основные особенности алгоритма эволюционной стратегии. Дайте схему канонического алгоритма эволюционной стратегии.
18. Перечислите основные операторы алгоритма эволюционного программирования. Представьте схему этого алгоритма.
19. Дайте схему алгоритма дифференциальной эволюции.
20. Представьте схему генетического коэволюционного алгоритма и раскройте суть его основных операторов.

АЛГОРИТМЫ РОЯ ЧАСТИЦ, КОЛОНИИ МУРАВЬЕВ И ПЧЕЛИНОГО РОЯ

Кроме рассмотренных в предыдущей главе эволюционных алгоритмов, из числа популяционных алгоритмов в настоящее время наиболее развитыми являются инспирированные живой природой алгоритмы роя частиц, муравьиной колонии и пчелиного роя. В данной главе эти алгоритмы рассмотрены в указанной последовательности.

3.1. ОПТИМИЗАЦИЯ РОЕМ ЧАСТИЦ

В основу *алгоритма оптимизации роем частиц* (*Particle Swarm Optimization, PSO*) положена социально-психологическая поведенческая модель толпы. Развитие алгоритма инспирировали такие задачи, как моделирование поведения птиц в стае и рыб в косяке. Целью было обнаружить базовые принципы, благодаря которым, например, птицы в стае ведут себя удивительно синхронно, меняя как по команде направления своего движения, так что стая движется как единое целое. К современному времени концепция алгоритма роя частиц развилась в высокоэффективный алгоритм оптимизации, часто составляющий конкуренцию лучшим модификациям генетического алгоритма.

Существует значительное число алгоритмов роя частиц. В каноническом алгоритме, предложенном в 1995 г. Кеннеди (*J. Kennedy*) и Эберхартом (*R. Eberhart*), при определении следующего положения частицы учитывается информация о наилучшей частице из числа ее «соседей», а также информация о данной частице на той итерации алгоритма, на которой этой частице соответствовало наилучшее значение фитнес-функции. Известно большое число модификаций данного алгоритма. Например, модификация *FIPS* (см. 3.1.2) при определении следующего положения частицы учитывает значения фитнес-функции, соответствующие всем частицам популяции.

Ниже представлены канонический алгоритм *PSO*, наиболее известные модификации канонического алгоритма, определено понятие топологии соседства частиц. Кроме того, рассмотрены основные топологии, используемые в вычислительной практике, дан обзор алгоритмов *PSO* с динамической топологией соседства частиц, представлен гибридный алгоритм на основе роя частиц и имитации отжига. В завершении параграфа приведен пример использования алгоритма *PSO*.

3.1.1. Канонический алгоритм роя частиц

Рассмотрим задачу многомерной глобальной безусловной минимизации. В каноническом *алгоритме PSO* координаты частицы s_i на итерации t определяет вектор $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,|X|})^T$, а на итерации $(t+1)$ – вектор $X'_i = (x'_{i,1}, x'_{i,2}, \dots, x'_{i,|X|})^T$. Начальные координаты частицы s_i равны $X_i(0)$; $i \in [1 : |S|]$.

пчела $s_*(t+1)$; пчелы множества $\tilde{S}^r = \{\tilde{s}_i^r, i \in [1:|\tilde{S}^r|]\}$, выполнившие вербовку; завербованные пчелы множества $S^f = \{s_i^f, i \in [1:|S^f|]\}$; пчелы числом $|S^o|$, случайные координаты которых получены по формуле вида (3.30). Вычисляем для всех указанных пчел полезность соответствующих решений.

Обновление динамических параметров алгоритма включает в себя следующие действия: увеличение значения счетчика числа итераций $t = t + 1$; понижение текущей «температуры» $\bar{\varepsilon}(t) = \alpha \bar{\varepsilon}(t)$, где $\alpha \in (0; 1]$ – коэффициент понижения «температуры»; уменьшение максимального отклонения

$$b_\delta = b_\delta \frac{\hat{t} - t}{\hat{t}}.$$

Завершение итераций предполагает проверку условий окончания итераций

$$t \geq \hat{t}, \bar{\varepsilon}(t) < \bar{\varepsilon}^{\min},$$

где $\bar{\varepsilon}^{\min}$ – минимально допустимая «температура». Если хотя бы одно из указанных условий выполнено, в качестве решения задачи принимаем решение $\tilde{X}^*(t)$ и завершаем вычисления, в противном случае переходим к шагу 2 рассмотренной схемы алгоритма.

Свободными являются следующие параметры алгоритма:

$\bar{\varepsilon}_0$ – начальная «температура»;

δ_ϕ – минимальный порог полезности пчелы (рекомендованное значение равно 0,1);

b_n – уровень возмущений (рекомендованное значение равно 0,1);

b_ϕ – вес компоненты $\bar{\phi}^r$ в привлекательности пчел;

b_r – порог привлекательности пчелы;

b_δ – максимально допустимое отклонение координат вербовавших и завербованных пчел;

α – коэффициент понижения текущей «температуры»;

$\bar{\varepsilon}^{\min}$ – минимально допустимая «температура».

Вопросы для самопроверки

1. Представьте схему канонического алгоритма роя частиц. Определите понятия *собственный путь* частицы и ее *локальный путь*.

2. Сформулируйте основные отличительные признаки следующих модификаций алгоритма роя частиц: притягательно-отталкивающий алгоритм, алгоритм роя частиц с управлением скоростью частиц, алгоритм роя частиц с полной информацией (*FIPS*), алгоритм *FIPS* с весами, ранжированный алгоритм *FIPS*, *FDR* алгоритм роя частиц.

3. Дайте определение понятия *топология соседства частиц*. Проиллюстрируйте топологии соседства типа клика, кольцо, двумерный тор, кластерная топология.

4. Определите понятие *динамическая топология соседства частиц*. Назовите основные особенности следующих алгоритмов роя частиц, использующих динамические топологии соседства: клубный алгоритм роя частиц, алгоритм роя частиц со стереотипированием, алгоритм с дополнением графа соседства частиц.

5. Каковы бионические предпосылки муравьиных алгоритмов оптимизации?

6. Приведите схему алгоритма непрерывной оптимизации колонией муравьев *CACO*. Детализируйте содержание основных этапов этого алгоритма.
7. Назовите основные отличительные признаки алгоритма непрерывно взаимодействующей муравьиной колонии *CIAC*.
8. Изложите схему непрерывного ортогонального алгоритма муравьиной колонии *COAC*.
9. Представьте схему гибридного алгоритма непрерывно взаимодействующей муравьиной колонии *HCIAC*.
10. Каковы бионические предпосылки алгоритмов оптимизации пчелиным роем?
11. Приведите схему пчелиного алгоритма (*Bees algorithm*).
12. Представьте схему алгоритма колонии искусственных пчел *ABC*. Детализируйте основные операторы этого алгоритма.
13. Дайте схему гибридного пчелиного алгоритма, использующего локальную оптимизацию путем имитации отжига.

ДРУГИЕ ПОПУЛЯЦИОННЫЕ АЛГОРИТМЫ, ВДОХНОВЛЕННЫЕ ЖИВОЙ ПРИРОДОЙ

Представляем алгоритмы, вдохновленные живой природой и менее разработанные по сравнению с аналогичными алгоритмами, рассмотренными в предыдущей главе. Алгоритмы упорядочены по убыванию степени их разработанности и широте применения.

4.1. ИСКУССТВЕННЫЕ ИММУННЫЕ СИСТЕМЫ

Искусственные иммунные системы (*Artificial Immune Systems, AIS*) представляют собой относительно новую область исследований. Первые попытки разработки *AIS* относятся к 70-м гг. XX века. Однако представляющие практический интерес результаты были получены только в 1990-х гг. В настоящее время *AIS* интенсивно развиваются и широко применяются в вычислительной практике. С 2002 г. ежегодно проводится международная конференция *International Conference on Artificial Immune Systems*, полностью посвященная данной области исследований.

AIS можно определить как информационную технологию, использующую понятия, аппарат и некоторые результаты теоретической иммунологии для решения прикладных задач. Кроме решения задач оптимизации, модели *AIS* широко используют для построения многоагентных и самоорганизующихся систем, моделей коллективного интеллекта и автономных распределенных систем, моделей искусственной жизни и обучающихся систем и тому подобное.

4.1.1. Биологические основы

Биологическим прототипом *AIS* является иммунная система человека и обработка информации в ней молекулами белков (пептидов). Эта система представляет собой сложную адаптивную структуру, использующую комбинацию различных механизмов защиты от внешних патогенов – любых микроорганизмов (включая вирусы и бактерии), способных вызывать патологическое состояние (болезнь) человека.

Защитная система человека имеет иерархическую многоуровневую структуру, при этом каждый следующий уровень повышает специфичность защиты. Первый эшелон защиты от патогенов реализует кожный покров, который создает физический барьер на пути бактерий и вирусов. Второй уровень защиты – биохимический. На этом уровне защиту обеспечивают пот, слюна и слезы. Третий и четвертый эшелоны защиты представляют врожденная и приобретенная иммунные системы. Если патоген проникает через два предыдущих барьера, то *неспецифическую реакцию (non-specific*

Оператор прививки применяют к агентам $s_i, s_j, i, j \in [1:|S|], i \neq j$, в том случае, когда мера $\mu(s_i, s_j)$ их близости (*similarity*) превышает порог $\varepsilon|X|$, где ε – заданная положительная константа (свободный параметр алгоритма). В качестве меры $\mu(s_i, s_j)$ в оригинальном алгоритме SSG использована величина

$$\mu(s_i, s_j) = \sum_{k=1}^{|X|} \frac{|x_{i,k} - x_{j,k}|}{x_k^+ - x_k^-}.$$

Прививку выполняем по правилу

$$x'_{i,k} = \begin{cases} x_{i,k}, & \frac{|x_{i,k} - x_{j,k}|}{x_k^+ - x_k^-} \leq \varepsilon|X|, \\ x_{j,k}, & \text{иначе} \end{cases}, \quad x'_{j,k} = \begin{cases} x_{j,k}, & \frac{|x_{i,k} - x_{j,k}|}{x_k^+ - x_k^-} \leq \varepsilon|X|, \\ x_{i,k}, & \text{иначе.} \end{cases}$$

Здесь $k \in [1:|X|]$. Другими словами, если модуль относительной разности между k -ми компонентами векторов X_i, X_j превосходит величину $\varepsilon|X|$, то эти вектора обмениваются k -ми компонентами.

После применения к текущей популяции указанных операторов получаем промежуточную популяцию $S \cup S'$, которой соответствует объединение исходных X_i и модифицированных векторов X'_i ; $i \in [1:|S|]$. После вычисления значений фитнес-функции во всех указанных точках из промежуточной популяции отбираем $|S|$ лучших агентов в популяцию S^{t+1} . Итерационный процесс повторяется до достижения заданного числа итераций.

Автором алгоритма SSG выполнено его исследование на ряде тестовых задач многомерной глобальной оптимизации, показавшее перспективность развития алгоритма.

Вопросы для самопроверки

1. Каковы биологические предпосылки искусственных иммунных систем?
2. Приведите общую схему решения задачи оптимизации с помощью модели иммунной сети.
3. Представьте схему алгоритма оптимизации на основе искусственной микроиммунной системы.
4. Сформулируйте биологические предпосылки бактериальной оптимизации.
5. Представьте схему канонического алгоритма бактериальной оптимизации.
6. Назовите принципы кооперативной бактериальной оптимизации и алгоритма, моделирующего эффект роения бактерий.
7. Приведите возможные схемы гибридизации бактериального алгоритма с другими популяционными алгоритмами.
8. Сформулируйте идеи, лежащие в основе алгоритма светлячков. Приведите схему этого алгоритма.
9. Назовите отличительные особенности алгоритма оптимизации роем светлячков.
10. Определите основные операторы алгоритма сорняковой оптимизации.
11. Представьте схему канонического алгоритма кукушкиного поиска, а также схему модифицированного алгоритма поиска кукушки.
12. Приведите схемы алгоритма обезьяньего поиска и обезьяньего алгоритма.
13. Назовите основные особенности тасующего алгоритма прыгающих лягушек, алгоритма, инспирированного летучими мышами, алгоритма поиска косяком рыб, алгоритма растущих деревьев.

Глава 5

ПОПУЛЯЦИОННЫЕ АЛГОРИТМЫ, ИНСПИРИРОВАННЫЕ НЕЖИВОЙ ПРИРОДОЙ, ЧЕЛОВЕЧЕСКИМ ОБЩЕСТВОМ, И ДРУГИЕ ПОПУЛЯЦИОННЫЕ АЛГОРИТМЫ

Материал главы можно разделить на три части. В первой части (параграфы 5.1–5.3) рассматриваем популяционные алгоритмы поисковой оптимизации, которые основаны на подражании неживой природе, – это алгоритмы поиска гармонии, гравитационного и электромагнитного поиска. Вторая часть (параграфы 5.4–5.8) включает в себя описание алгоритмов, инспирированных человеческим обществом, – алгоритмы эволюции разума и стохастического диффузионного поиска, культурный и миметические алгоритмы, а также самоорганизующийся миграционный алгоритм. Последняя часть (параграф 5.9) представлена алгоритмами рассеянного поиска и прокладки путей, которые трудно отнести к перечисленным выше классам алгоритмов.

Мы не рассматриваем широко известный *алгоритм интеллектуальных капель воды* (*Intelligent Water Drop algorithm, IWD*), предложенный Шах-Хосейни (*H. Shah-Hosseini*) в 2009 г. Алгоритм *IWD* вдохновлен способностью рек прокладывать близкие к оптимальным пути своего течения и предназначен для решения задач на графах, таких как задача о коммивояжере (*Travelling Salesman Problem, TSP*) или задача экономического распределения грузовиков (*Economic Load Dispatch Problem, ELDP*). По той же причине в главе не нашел места алгоритм *RFD* (*River Formation Dynamics*), предложенный Рабаналом (*P. Rabanal*) с соавторами в 2007 г.

5.1. ГАРМОНИЧЕСКИЙ ПОИСК

Алгоритм поиска гармонии (*Harmony Search algorithm, HS*) предложен в 2001 г. Гимом (*Z. W. Geem*). Алгоритм вдохновлен процессом поиска музыкантами гармонии в музыке. Ситуации идеальной гармонии звуков алгоритм *HS* ставит в соответствие глобальный экстремум в задаче многомерной оптимизации, а процессу импровизации музыканта – процедуру поиска этого экстремума.

В процессе импровизации каждый музыкант на каком-либо такте музыкального произведения воспроизводит звук (в рамках возможностей его музыкального инструмента), так что звуки всех музыкантов оркестра на этом такте образуют один вектор гармонии. Векторы, образующие «хорошие» гармонии, хранятся в памяти каждого из музыкантов и могут быть ими использованы для формирования «хороших» гармоний на следующих тактах произведения. Как правило, в процессе импровизации музыкант выполняет одно из трех следующих требований: сформировать абсолютно случайный звук из доступного ему диапазона звуков; воспроизвести какой-либо звук из своей памяти гармоний; сыграть смежный вектор гармонии из той же памяти.

ритм прокладки путей назад (*backward PR*) используют в качестве инициализирующего решения худшее и лучшее решение из числа X' , X'' соответственно. Комбинированный алгоритм прокладки путей (*back and forward PR*) строит две различные траектории, первая из которых исходит из решения X' , а вторая – из решения X'' . Смешанный PR-алгоритм (*mixed PR*) прокладывает два пути, идущих из решений X' , X'' , до тех пор, пока эти пути не образуют решение, равноудаленное от X' , X'' , то есть пока указанные пути не встретятся. Рандомизированный PR-алгоритм (*randomized PR*) в качестве следующего за X решения на пути $P(X', X'')$ выбирает не лучшее решение из множества $d_T(X)$, а случайное решение. Наконец, усеченный PR-алгоритм (*truncated PR*) исследует не всю траекторию $P(X', X'')$, а лишь некоторую ее часть.

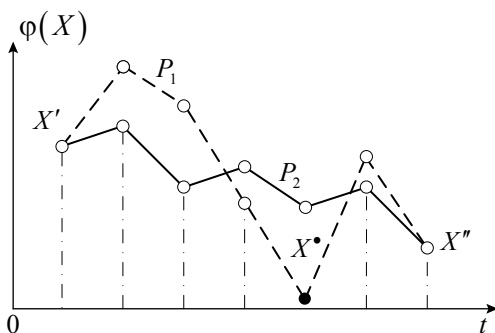


Рис. 5.22. К формированию траекторий в алгоритме PR

Вопросы для самопроверки

1. Представьте схему канонического алгоритма поиска гармонии. Определите основные операторы этого алгоритма.
2. Сформулируйте возможные направления модификации канонического алгоритма поиска гармонии.
3. Дайте схему детерминированного алгоритма оптимизации центральной силой, а также схему алгоритма гравитационного поиска как развитие первого из указанных алгоритмов.
4. Назовите основные подходы к модификации алгоритма гравитационного поиска.
5. Приведите схему канонического электромагнитного алгоритма оптимизации. Определите основные операторы этого алгоритма, а также возможные направления его модификации.
6. Представьте схему алгоритма поиска системой зарядов.
7. Дайте схему простого алгоритма эволюции разума. Определите основные операторы алгоритма.
8. Сформулируйте основные отличительные признаки расширенного, улучшенного, а также хаотического алгоритмов эволюции разума.
9. Назовите предпосылки алгоритма стохастического диффузионного поиска, представьте схему алгоритма и детализацию его основных операторов.
10. Определите основные понятия культурного алгоритма, приведите его общую схему, а также примеры операторов, обеспечивающих построение и коррекцию пространства убеждений.
11. Приведите примеры культурной модификации генетических операторов.
12. Дайте определение меметических алгоритмов оптимизации. Приведите общую схему меметического алгоритма. Определите основные гиперэвристики, используемые в гиперэвристических адаптивных мультимемных алгоритмах. Сформулируйте основные принципы построения самоадаптирующихся мультимемных алгоритмов.
13. Представьте схему самоорганизующегося миграционного алгоритма.
14. Сформулируйте основные отличительные черты алгоритмов рассеянного поиска и прокладки путей. Приведите схемы этих алгоритмов.

Глава 6

ГИБРИДИЗАЦИЯ ПОПУЛЯЦИОННЫХ АЛГОРИТМОВ

Опыт решения сложных прикладных задач, сводящихся к задаче глобальной оптимизации либо включающих в себя задачу глобальной оптимизации, показывает, что применение любого одного алгоритма оптимизации (как классического, так и популяционного) далеко не всегда приводит к успеху. В гибридных (комбинированных) алгоритмах, объединяющих различные либо одинаковые алгоритмы, но с различными значениями свободных параметров, преимущества одного алгоритма могут компенсировать недостатки другого. Поэтому одним из основных путей повышения эффективности решения задач глобального поиска в настоящее время является разработка гибридных популяционных алгоритмов. Иногда методы комбинирования метаэвристических алгоритмов называют *гиперэвристиками* (*hyper-heuristics*).

В этой главе представлены общие принципы гибридизации популяционных поисковых алгоритмов и рассмотрены основные методы гибридизации.

6.1. ОБЩИЕ ПРИНЦИПЫ ГИБРИДИЗАЦИИ

Можно предложить значительное число способов гибридизации поисковых алгоритмов оптимизации вообще и популяционных алгоритмов в частности, так что даже существует самостоятельная проблема классификации этих способов. Рассмотрим три следующие классификации: одноуровневая классификация Ванга (*X. Wang*), двухуровневая классификация Эль-Абда и Камэла (*El-Abd, M. Kamel*), четырехуровневая классификация Рейдла (*G. R. Raidl*).

6.1.1. Одноуровневая классификация Ванга

В соответствии с классификацией Ванга выделяют три категории гибридных алгоритмов: вложенные (*embedded*) алгоритмы, алгоритмы типа препроцессор/постпроцессор (*preprocessor/postprocessor*), коалгоритмы (*co-algorithms*).

В категории методов *гибридизации вложением* выделяем высокоуровневую и низкоуровневую гибридизации, как это сделано в четырехуровневой классификации Рейдла (см. ниже).

Высокоуровневая гибридизация вложением (*high-level embedded hybridization*) предполагает слабую связь объединяемых алгоритмов, обычно основанную на комбинировании популяционного алгоритма глобального поиска и того или иного алгоритма локального поиска (не обязательно популяционного). Отличительным признаком высокоуровневой гибридизации является то, что объединяемые алгоритмы сохраняют значительную автономию, так что

к выбору значений их свободных параметров, а во втором – к выбору типа алгоритма (и, быть может, значений его параметров). И в том, и в другом случае можно использовать альтернативную адаптацию или метаоптимизационные подходы, рассматриваемые в гл. 7.

Вопросы для самопроверки

1. Назовите классификационные признаки способов гибридизации популяционных алгоритмов в одноуровневой классификации Ванга, двухуровневой классификации Эль-Абда и Камэла и четырехуровневой классификации Рейдла.
2. Сформулируйте основные способы высокоуровневой и низкоуровневой гибридизации вложением.
3. Определите основные способы гибридизации по схеме препроцессор/постпроцессор.
4. Сформулируйте принципы коэволюционного подхода к решению задачи оптимизации. Определите используемые в коэволюционных алгоритмах модели коэволюции.
5. Дайте схему гибридного коалгоритма на основе алгоритма роя частиц.

Глава 7

МЕТАОПТИМИЗАЦИЯ ПОПУЛЯЦИОННЫХ АЛГОРИТМОВ

Назовем вектор свободных параметров популяционного алгоритма оптимизации *стратегией алгоритма*. Стратегия популяционного алгоритма в значительной степени определяет его эффективность. Поэтому возникает задача поиска оптимальной стратегии алгоритма, которая называется *метазадачей оптимизации*. Алгоритм оптимизации и соответствующую программу назовем *базовым алгоритмом* и *базовой программой*. Аналогично метод, алгоритм и программу решения метазадачи назовем *метаметодом*, *метаалгоритмом* и *метапрограммой*.

Далее представляем постановку метазадачи и классификацию методов ее решения, рассматриваем несколько подходов к решению задачи параметрической метаоптимизации. В завершение обсуждаем задачу структурной метаоптимизации.

7.1. ПОСТАНОВКА МЕТАЗАДАЧИ ОПТИМИЗАЦИИ

Некоторый класс задач многомерной глобальной безусловной минимизации обозначим $\Phi = (\phi_i, i = 1, 2, \dots)$, где $\phi_i = \phi$ одна из задач этого класса. Отметим, что поскольку речь идет о безусловной оптимизации, задачу $\phi_i = \phi$ полностью определяет минимизируемая функция $f_i = f$.

7.1.1. Статическая параметрическая метаоптимизация

Статическая параметрическая метазадача (*parameter adaptation problem*) предполагает, что оптимальная стратегия базового алгоритма является статической и отыскивается для класса задач Φ .

Пусть $a(B)$ – рассматриваемый базовый алгоритм решения задач класса Φ , где $B = (b_1, b_2, \dots, b_{|B|})$ – вектор стратегий этого алгоритма. Положим, что для каждого из компонент b_i вектора B заданы интервалы его допустимых значений b_i^-, b_i^+ , так что определено множество D_B допустимых стратегий

$$D_B = \{b_i \mid b_i^- \leq b_i \leq b_i^+, i \in [1 : |B|]\}.$$

Совокупность алгоритмов, которую определяет множество D_B , обозначим

$$A = \{a(B) \mid B \in D_B\}.$$

Рассматриваем две параметрические метазадачи, первая из которых ставится следующим образом. Для функции $\phi \in \Phi$ найти стратегию B^* (а тем самым, и алгоритм $a^* = a(B^*) \in A$), которая обеспечивает минимальное значение *целевой метафункции* $\mu(\phi, B)$:

$$\min_{B \in D_B} \mu(\phi, B) = \mu(\phi, B^*). \quad (7.1)$$

ных операторов. Вычислительная сложность различных вариантов операторов может весьма существенно отличаться. Пусть поэтому вторая метафункция $\mu_2(\phi, O)$ представляет собой число арифметических операций, реализующих рассматриваемую комбинацию этих вариантов.

Решение задачи (7.10) может быть выполнено методом адаптации или самоадаптации на множестве стратегий D_O .

Метод *адаптации*. В простейшем случае адаптационную стратегию можно построить с помощью метода рулетки по следующей схеме.

1) Поставим в соответствие каждой из $|D_O|$ допустимых комбинаций O_i рассматриваемых вариантов операторов некоторую начальную вероятность их применения ξ_i , так что имеет место равенство

$$\sum_{i \in [1:|D_O|]} \xi_i = 1.$$

2) По правилу рулетки в соответствии с вероятностью ξ_i выбираем комбинацию O_i .

3) В основном цикле генетического алгоритма определяем значения величин $\mu_1(\phi, O_i)$, $\mu_2(\phi, O_i)$, а также их скалярной свертки $\mu(\phi, O_i)$.

4) Изменяем распределение вероятностей ξ_i так, чтобы вероятность ξ_i была обратно пропорциональна значению целевой метафункции $\mu(\phi, O_i)$, и продолжаем итерационный процесс путем перехода к шагу 2.

5) Вычисления заканчиваем при выполнении условий окончания итераций.

Очевидным вариантом такого подхода к решению метазадачи динамической структурной оптимизации является вариант, в котором по рассмотренной схеме задают вероятности применения не наборов O_i , а непосредственно каждого из допустимых операторов $o_{i,j}$.

Метод *самоадаптации*. Введем в рассмотрение $((|X|+1) \times 1)$ -вектор

$$\bar{X} = (X, n_o) = (x_1, x_2, \dots, x_{|X|}, n_o),$$

где $n_o \in [1:|D_O|]$ – номер набора операторов из множества D_O . Экзон мультихромосомы \bar{H} в этом случае представляет собой вектор варьируемых параметров базовой задачи, а интрон – номер набора n_o . Указанные части мультихромосомы \bar{H} (экзон и интрон) эволюционируют в соответствии с функциями приспособленности $\phi(X)$, $\mu(\phi, B)$ соответственно.

Генетические операторы, применяемые к экзону и интрону, различны. В первом случае речь идет о непрерывных генетических операторах, а во втором случае – о дискретных операторах.

Вопросы для самопроверки

1. Приведите постановку статической задачи параметрической метаоптимизации. Представьте общие постановки задач неадаптивной и адаптивной динамической метаоптимизации.

2. Приведите постановки задач структурной и структурно-параметрической метаоптимизации.

3. Перечислите основные классификационные признаки методов метаоптимизации по Эйбену и Смиту – Фогарту.

4. Изложите суть следующих методов параметрической метаоптимизации: однократная настройка параметров, перманентная настройка параметров, адаптивное управление параметрами, самоадаптивное управление параметрами.
5. Назовите возможные объекты, уровни и основания метаоптимизации.
6. Дайте общие схемы статической параметрической метаоптимизации, динамической неадаптивной и динамической адаптивной метаоптимизации.
7. Изложите схему перманентной настройки параметров.
8. Раскройте суть гомогенных и гетерогенных коэволюционных методов адаптивного управления параметрами.
9. Приведите схемы методов самоадаптивного управления параметрами на основе расширения вектора варьируемых параметров исходной задачи, а также на основе коэволюционного подхода.
10. Определите схемы статической и динамической структурной метаоптимизации.

Глава 8

ПОПУЛЯЦИОННЫЕ АЛГОРИТМЫ МНОГОЦЕЛЕВОЙ ОПТИМИЗАЦИИ

В русскоязычной литературе задачу поиска экстремумов одновременно нескольких функций обычно называют задачей многокритериальной оптимизации. Точнее будет использовать термин *многоцелевая оптимизация*.

Классические алгоритмы решения задачи многоцелевой оптимизации (МЦО-задачи) основаны на сведении ее к совокупности задач глобальной одноцелевой оптимизации. Относительно новый и быстро развивающийся класс алгоритмов многоцелевой оптимизации образуют *алгоритмы Парето-аппроксимации*, предполагающие предварительное построение некоторой конечномерной аппроксимации множества Парето, а тем самым и фронта Парето.

Обычно алгоритмы Парето-аппроксимации строят на основе эволюционных и, чаще всего, генетических алгоритмов. Такие алгоритмы Парето-аппроксимации называют эволюционными. Принципиальными в этом случае являются правила формирования фитнес-функции, обеспечивающей в конечном счете перемещение индивидов популяции в направлении множества Парето. Эволюция же этих индивидов может протекать по законам не только эволюционных, но и других популяционных алгоритмов. Поэтому в качестве общего названия рассматриваемых в данной главе алгоритмов используем термин *популяционные алгоритмы Парето-аппроксимации*.

В первом параграфе приведена постановка МЦО-задачи, даны определения множества и фронта Парето, классификация алгоритмов решения задачи. Для полноты картины сначала рассматриваем наиболее известные непопуляционные алгоритмы Парето-аппроксимации (второй параграф). Обзор популяционных алгоритмов Парето-аппроксимации представлен в третьем параграфе. При этом внимание сконцентрировано преимущественно на правилах формирования фитнес-функций, используемых в этих алгоритмах. Самостоятельную проблему в задаче Парето-аппроксимации представляет разработка критериев оценки качества получаемой аппроксимации. Основные из этих критериев рассматриваем в четвертом параграфе. Пятый параграф посвящен методам обеспечения качества Парето-аппроксимации. Последний (шестой) параграф представляет примеры решения задачи Парето-аппроксимации с помощью генетического алгоритма, а также с помощью алгоритмов роя частиц, колонии муравьев и колонии пчел.

8.1. ЗАДАЧА МНОГОЦЕЛЕВОЙ ОПТИМИЗАЦИИ (МЦО-ЗАДАЧА) И АЛГОРИТМЫ ЕЕ РЕШЕНИЯ

В данном разделе приводим постановку МЦО-задачи, даем определение множества и фронта Парето, формулируем основные свойства этих объектов, представляем классификацию алгоритмов решения МЦО-задачи.

Используем следующие значения свободных параметров пчелиного алгоритма: $|S^o| = 200$; $|A^b| = 40$; $n^b = 60$; $\hat{t} = 100$. При этом получено 55 недоминируемых решений (рис. 8.40).

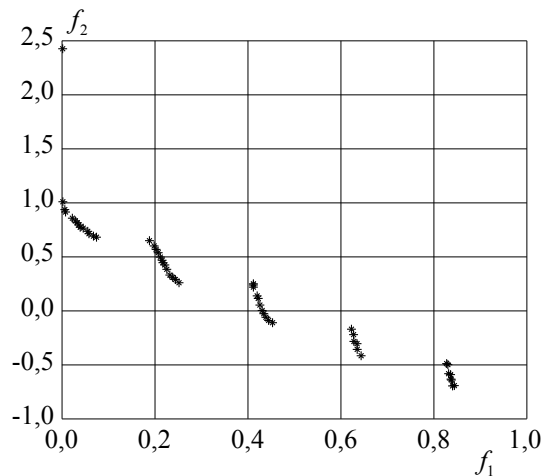


Рис. 8.40. Аппроксимация фронта Парето, задача (8.22), (8.23); модификация пчелиного алгоритма: $\hat{t} = 100$

Те же 100 итераций алгоритма *MOPSO* позволили найти 48 недоминируемых решений задачи (8.22), (8.23). Таким образом, и для сложной МЦО-задачи, имеющей множественные разрывы фронта Парето, рассматриваемая модификация пчелиного алгоритма показывает более высокую эффективность в сравнении с алгоритмом *MOPSO*.

Вопросы для самопроверки

1. Приведите постановку задачи многоцелевой оптимизации. Определите понятия: множество достижимости, отношение доминирования, отношение строгого доминирования, фронт Парето, множество Парето, Парето-аппроксимация.
2. Дайте общие схемы решения задачи многоцелевой оптимизации с помощью алгоритма зондирования, а также априорных, апостериорных и адаптивных алгоритмов.
3. Представьте схемы основных непопуляционных алгоритмов Парето-аппроксимации.
4. Опишите популяционные алгоритмы Парето-аппроксимации на основе лексикографической турнирной селекции.
5. Приведите схемы основных популяционных алгоритмов Парето-аппроксимации на основе чередующихся целевых функций.
6. Представьте схемы наиболее известных популяционных алгоритмов Парето-аппроксимации, использующих ранжирование агентов популяции.
7. Дайте схемы некоторых популяционных алгоритмов Парето-аппроксимации, не использующих ранжирование агентов популяции.
8. Назовите основные унарные и бинарные критерии оценки качества Парето-аппроксимации.
9. Перечислите основные средства обеспечения разнообразия популяции.
10. Определите метод обеспечения разнообразия популяции путем нишевания.

ПАРАЛЛЕЛЬНЫЕ ПОПУЛЯЦИОННЫЕ АЛГОРИТМЫ ПОИСКОВОЙ ОПТИМИЗАЦИИ

Популяционные алгоритмы поисковой оптимизации могут требовать больших вычислительных ресурсов даже при невысокой вычислительной сложности целевой функции (целевых функций) в задаче одноцелевой (многоцелевой) оптимизации. Например, обычной является ситуация, когда решение задачи оптимизации требует сотен тысяч и миллионов вычислений значений этой (этих) функций. С другой стороны, в практически значимых задачах одно испытание может требовать нескольких часов работы современной однопроцессорной ЭВМ даже в том случае, когда этот процессор является многоядерным. В силу данных обстоятельств весьма привлекательным выглядит использование параллельных ЭВМ для ускорения оптимизационных вычислений.

Некоторые параллельные алгоритмы сами по себе могут оказывать положительный эффект на процесс оптимизации. Так, Сколички (*Z. Skolicki*) показал, что для некоторых целевых функций параллельный алгоритм на основе островной модели параллелизма (см. ниже) более эффективен, чем соответствующий последовательный алгоритм, даже если не учитывать ускорение, полученное в результате использования параллельной ЭВМ.

Многие непопуляционные алгоритмы стохастической оптимизации могут быть распараллелены, но, как правило, весьма искусственным образом. Важным достоинством популяционных алгоритмов является их естественный параллелизм, поскольку в этих алгоритмах значительное число агентов на логическом уровне эволюционирует параллельно.

В разработку и развитие параллельных популяционных алгоритмов оптимизации внесли свой вклад многие исследователи, среди которых особо следует отметить Грефенстетта (*J. Grefenstette*). Технический отчет, выпущенный им в свет в 1981 г., предвосхитил очень многое в теории и практике этих алгоритмов.

Представляем классификацию и основные типы параллельных ЭВМ (первый параграф). Приводим постановку задачи балансировки загрузки этих ЭВМ, а также основные методы ее решения (второй параграф). Даем обзор известных методов распараллеливания, на основе которых могут быть построены параллельные популяционные алгоритмы (третий параграф). Рассматриваем примеры параллельных популяционных алгоритмов одно- и многоцелевой оптимизации (четвертый параграф).

9.1. КЛАССИФИКАЦИЯ И ОСНОВНЫЕ ТИПЫ ПАРАЛЛЕЛЬНЫХ ЭВМ

Нестрого *архитектуру* вычислительной системы можно определить как абстрактное представление ЭВМ, какой ее видит программист, занимающийся низкоуровневым программированием. Полное описание архитектуры сис-

Вопросы для самопроверки

1. Представьте основные классы параллельных вычислительных систем в соответствии с классификацией Флинна, а также с классификациями по типу строения оперативной памяти, по типу коммуникационной сети, по степени однородности.
2. Назовите основные архитектурные особенности следующих классов параллельных вычислительных систем: мультипроцессоры, мультимикропроцессоры, графические процессорные устройства.
3. Представьте постановки задачи балансировки загрузки в терминах математического программирования и терминах теории графов.
4. Опишите метод статической балансировки загрузки, использующий алгоритм упаковки в контейнеры, а также иерархический графовый метод.
5. Приведите описания следующих методов динамической балансировки загрузки: экспоненциальная балансировка, диффузная балансировка.
6. Перечислите основные классы моделей параллелизма.
7. Определите общую схему параллельных популяционных алгоритмов оптимизации, использующих глобальную модель параллелизма.
8. Изложите схему параллельных популяционных алгоритмов оптимизации, использующих островную и диффузную модели параллелизма.
9. Назовите основные особенности следующих моделей параллелизма: модель некоммутирующей мультипопуляции, пространственно вложенная модель, модель расщепления, модель коэволюции, комбинированная модель.

Литература

1. Аттетков А. В., Галкин С. В., Зарубин В. С. Методы оптимизации: Учебник для ВУЗов. – М.: Из-во МГТУ им. Н. Э. Баумана, 2001, 439 с.
2. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002, 608 с.
3. Курейчик В. В., Курейчик В. М., Родзин С. И. Теория эволюционных вычислений. М.: ФИЗМАТЛИТ, 2012, 260 с.
4. Ларичев О. И. Теория и методы принятия решений: Учебник для ВУЗов. М.: Университетская книга, Логос, 2006, 392 с.
5. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. / Рутковская Д., Пилиньский М., Рутковский Л. М.: Горячая линия–Телеком, 2004, 452 с.
6. Семенкин Е. С. Эволюционные методы моделирования и оптимизации сложных систем / Семенкин Е. С. и др. // Электронные публикации НГУ. (http://library.krasu.ru/ft/ft/_umkd/22/u_lectures.pdf)
7. Черноуцкий И. Г. Методы принятия решений: Учебное пособие. СПб.: БХВ-Петербург, 2005, 416 с.
8. Штойер Р. Многокритериальная оптимизация. М.: Радио и связь, 1992, 504 с.
9. Blum Ch., Aguilera M. J. B., Roli A., Sampels M. (Eds.) Hybrid Metaheuristics. An Emerging Approach to Optimization. Springer-Verlag, Berlin, Germany, 2008, 289 pp.
10. Deb K. Multi-objective optimization using evolutionary algorithms. Chichester, UK: Wiley, 2001, 518 pp.

Дополнительная литература

1. Гладков Л. А., Курейчик В. В., Курейчик В. М., Сороколетов П. В. Биоинспирированные методы в оптимизации. М.: ФИЗМАТЛИТ, 2009, 384 с.
2. Семенкин Е. С., Жукова М. Н., Жуков В. Г., Панфилов И. А., Тынченко В. В. Эволюционные методы моделирования и оптимизации сложных систем. Конспект лекций. Красноярск, 2007, 515 с. (http://files.lib.sfu-kras.ru/ebibl/umkd/22/u_lectures.pdf)
3. Ashlock D. Evolutionary Computation for Modeling and Optimization. Springer-Verlag, Berlin, Germany, 2006, 571 pp.
4. Blum C., Andrea R., Michael S. (Eds.) Hybrid Metaheuristics. An Emerging Approach to Optimization. Springer-Verlag, Berlin, Germany, 2008, Vol. 114, 290 p.
5. Dos Santos W. P. (ed.). Evolutionary Computation. InTech, 2009, 582 pp.
6. Engelbrecht A. P. Computational Intelligence. An Introduction. Second Edition. John Wiley & Sons, 2007, 597 pp.
7. Kita E. (ed.). Evolutionary Algorithms. InTech, 2011, 596 pp.
8. Luke S. Essentials of Metaheuristics, Lulu, 2009. (<http://cs.gmu.edu/~sean/book/metaheuristics>)
9. Pohlheim H. Evolutionary Algorithms: Overview, Methods and Operators. (<http://www.geatbx.com/>)
10. Yang X. S. Nature-inspired Metaheuristic Algorithms. Luniver Press, 2010, UK, pp. 81–95.

11. Suganthan P.N., Hansen N., Liang J.J., Deb K., Chen Y.P., Auger A., Tiwari S. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical Report / Nanyang Technological University, Singapore, 2005 (<http://www.ntu.edu.sg/home/EPNSugan>)
12. Tang K., Yao X., Suganthan P.N., MacNish C., Chen Y.P., Chen C.M., Yang Z. Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization. Technical Report / Nature Inspired Computation and Applications Laboratory, USTC, China, 2007. (<http://nical.ustc.edu.cn/cec08ss.php>)
13. Zhang Q., Zhou A., Zhao Sh., Suganthan P.N., Liu W., Tiwari S. Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. Technical Report CES-487 / The School of Computer Science and Electronic Engineering University of Essex, Colchester, C04, 3SQ, UK, 2009, 30 pp.
14. Zitzler E., Deb K., Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results // *Evolutionary Computation*, 2000, Vol. 8 (2), pp. 173–195.

Предметный указатель

Алгоритм		– непрерывный (<i>CACO</i>)	3.2.2
– адаптивный взвешенных сумм	8.2.2	– гибридный (<i>HCIAC</i>)	3.2.5
– бактериальной оптимизации	4.2	– динамический (<i>DHCIAC</i>)	3.2.5
– адаптивный (<i>ABFO</i>)	4.2.3	– ортогональный (<i>COAC</i>)	3.2.4
– использующий эффект роения бактерий	4.2.4	– коммутирующей мульти-популяции	9.4.2
– канонический (<i>BFO</i>)	4.2.2	– комплексов	1.2.3
– кооперативный (<i>CBFO</i>)	4.2.3	– модифицированный	1.3.2
– гибридный (<i>CBFO-H</i>)	4.2.3	– композитных точек	8.3.3
– последовательный (<i>CBFO-S</i>)	4.2.3	– конфигураций	1.2.3
– взвешенных целевых функций	8.2.2	– кукушкиного поиска (<i>CS</i>)	4.5
– восхождения на гору	2.7	– улучшенный (<i>ICS</i>)	4.5
– генетический		– модифицированный (<i>MCS</i>)	4.5
– адаптивный	2.7	– культурный	5.6
– бинарный	2.2	– лексикографического упорядочения	8.1.2
– гибридный	2.7	– лексикографической турнирной селекции	8.3.1
– канонический	2.7	– локальной унимодальной выборки (<i>LUS</i>)	1.2.1
– клеточный	2.7, 9.4.3	– локальный хаотический (<i>CLS</i>)	1.4.2
– коэволюционный	2.12	– мемеевый	5.7
– многопопуляционный	2.7	– адаптивный	5.7.2
– непрерывный	2.2	– самоадаптирующийся	5.7.3
– островной	2.7	– микроиммунной системы (<i>Micro-AIS</i>)	4.1.3
– простой	2.7	– миграционный	9.4.2
– сегрегированный	2.6	– Монте-Карло	1.4.2
– <i>CHC</i>	2.7	– мультимемный	5.7.1
– <i>Genitor</i>	2.7	– мультистарта	1.4.2
– <i>NPGA</i>	8.3.3	– недоминируемой сортировки (<i>NDS</i>)	8.3.3
– генетического программирования	7.3.2	– обезьяний	4.6.2
– гиперкубовый	8.3.4	– обезьяньего поиска (<i>MS</i>)	4.6.1
– гиперсферы	1.2.3	– одномерного рандомизированного поиска (<i>SDPS</i>)	1.2.1
– динамического симплекса	3.2.5	– одномерных вложений	1.4.1
– динамического соседства	8.3.2	– Парето-	
– дифференциальной эволюции	2.11	– аппроксимации	8
– зондирования	8.1.2	– силы	8.3.3
– идеальной точки	8.2.2	– переключающихся целевых функций	8.3.2
– имитации отжига (<i>SA</i>)	1.4.2	– поиска	
– инспирированный летучими мышами (<i>BI</i>)	4.7.2	– гармонии (<i>HS</i>)	
– искусственной иммунной системы	4.1	– глобально лучший (<i>GHS</i>)	5.1.2
– клонального отбора (<i>CLONALG</i>)	4.1.3	– дифференциальный (<i>DHS</i>)	5.1.2
– колонии искусственных пчел (<i>ABC</i>)	3.3.3	– канонический	5.1.1
– колонии муравьев (<i>ACO</i>)	3.2		
– непрерывно взаимодействующей (<i>CIAC</i>)	3.2.3		

– улучшенный (<i>IHS</i>)	5.1.2	– с запретами (<i>TS</i>)	1.4.3
– гравитационный (<i>GS</i>)	5.2	– с чередующимися окрестностями (<i>VNS</i>)	1.4.2
– косяком рыб (<i>FSS</i>)	4.7.3	– самоорганизующийся миграционный (<i>SOM</i>)	5.8
– плотный (<i>DFSS</i>)	4.7.3	– сверхбыстрого отжига	1.4.2
– рассеянный (<i>SS</i>)	5.9.1	– светлячковый	4.3.1
– с чередующимися окрестностями	1.4.2	– роевый (<i>GSO</i>)	4.3.2
– системой зарядов (<i>CSS</i>)	5.3.2	– случайного поиска (<i>RS</i>)	1.2.1
– со случайно чередующимися окрестностями (<i>RVNS</i>)	1.4.2	– адаптивный (<i>ASSRS</i>)	1.2.1
– стохастического диффузионного (<i>SDS</i>)	5.5	– адаптивный максимизирующий (<i>AMRS</i>)	1.2.2
– контекстно-зависимый	5.5	– жадный (<i>GRASP</i>)	1.4.2
– фокусированный	5.5	– неадаптивный (<i>FSSRS</i>)	1.2.1
– пороговый	1.4.2	– с интенсификацией и диверсификацией (<i>ARSID</i>)	1.2.1
– последовательных уступок	8.1.2	– с постоянным радиусом поиска и случайными направлениями	1.2.3
– прокладки путей (<i>PR</i>)	5.9.2	– сорняковой оптимизации (<i>IWO</i>)	4.4
– вперед	5.9.2	– стохастический линейный (<i>RLS</i>)	1.2.1
– комбинированный	5.9.2	– ранжирования	
– многородительский	5.9.2	– средневзвешенного (<i>WAR</i>)	8.3.3
– назад	5.9.2	– максимального взвешенного (<i>WMR</i>)	8.3.3
– периодический	5.9.2	– стохастической аппроксимации (<i>SPSA</i>)	1.2.1
– рандомизированный	5.9.2	– суммы взвешенных оценок (<i>SWR</i>)	8.3.3
– смешанный	5.9.2	– глобальных (<i>SWGR</i>)	8.3.3
– усеченный	5.9.2	– с эффектом Болдуина	2.7
– прыгающих лягушек тасующий (<i>SFL</i>)	4.7.1	– «хищник–жертва»	8.3.2
– пчелиного роя		– центральной силы	5.2
– базовый (<i>BA</i>)	3.3.2	– чередующихся целевых функций	8.3.2
– улучшенный (<i>IBA</i>)	3.3.2	– штрафных функций	1.3.1
– растущих деревьев (<i>SSG</i>)	4.7.4	– эволюции разума (<i>MEC</i>)	5.4
– роя частиц (<i>PSO</i>)	3.1	– простой (<i>SMEC</i>)	5.4.1
– волевой (<i>VPSO</i>)	4.7.3	– расширенный (<i>EMEC</i>)	5.4.2
– канонический	3.1.1	– улучшенный (<i>IMEC</i>)	5.4.3
– клубный (<i>CBPSO</i>)	3.1.4	– хаотический (<i>CMEC</i>)	5.4.4
– многоцелевой (<i>MOPSO</i>)	8.6.2, 9.5.2	– эвристический на основе паттернов (<i>HPS</i>)	1.2.1
– многороевый (<i>MS MOPSO</i>)	8.6.2	– электромагнитный (<i>EM</i>)	
– притягательно-отталкивающий (<i>ARPSO</i>)	3.1.2	– канонический	5.3.1
– ранжированный (<i>RFIPS</i>)	3.1.2	– <i>FFGA</i>	8.3.3
– с весами (<i>WFIPS</i>)	3.1.2	– <i>k</i> -средних	7.4.2
– с дополнением графа соседства (<i>DSPSO</i>)	3.1.4	– <i>MOGA</i>	8.3.3
– со стереотипированием	3.1.4	– <i>NSGA</i>	8.3.3
– с полной информацией (<i>FIPS</i>)	3.1.2	– <i>NSGA II</i>	8.3.3
– с управлением скоростью частиц	3.1.2	– <i>PAES</i>	8.3.3
– <i>ARPSO</i>	3.1.2	– <i>SPEA</i>	8.3.3
– <i>FDR</i>	3.1.2	– <i>SPEA 2</i>	8.3.3
– с возвратом при неудачном шаге	1.2.1	– <i>VEGA</i>	8.3.2

Аппроксимация направления		– сходимости	
убывания функции (<i>Approximate Descent Direction, ADD</i>)	1.2.1	– сильный	5.5
		– слабый	5.5
Аутбридинг	2.5.2	– унарный	8.4
Гибридизация		Кроссовер	2.1
– вложением	6.1.1	– арифметический	2.4.2
– высокоуровневая	6.1.1	– равномерный	2.4.2
– последовательная	6.2.1	– бинарный	2.4.1
– на основе декомпозиции пространства поиска	6.2.1	– вещественный	2.4.2
– низкоуровневая	6.1.1, 6.2.2	– геометрический	2.4.2
– высокого уровня	6.1.3	– равномерный	2.4.2
– коалгоритмическая	6.1.1	– двухродительский	2.4.1
– низкого уровня	6.1.3	– двухточечный	2.4.1
– параллельная	6.1.3	– дискретный	2.4.2
– последовательная	6.1.3	– имитирующий бинарный (<i>SBX</i>)	2.4.2
– типа препроцессор / пост-процессор	6.1.1, 6.3	– комбинированный	2.4.1
– конвейерная	6.1.1, 6.3.2	– линейный	2.4.2
– последовательная	6.1.1, 6.3.1	– расширенный	2.4.2
– чередующаяся	6.1.3	– многогородительский	2.4.1
Гиперэвристика	5.7.1	– многоточечный	2.4.1
– жадная	5.7.2	– нечеткий	2.4.2
– случайная	5.7.2	– однородный	2.4.1
– с функцией выбора	5.7.2	– одноточечный	2.4.1
Декомпозиция пространства поиска		– перетасовочный	2.4.1
– комбинированная	6.1.2	– плоский	2.4.2
– неявная	6.1.2	– простейший	2.4.2
– явная	6.1.2	– смешанный	2.4.2
Инбридинг	2.5.2	– равномерный	2.4.2
Интрон	7.6.1	– триадный	2.4.1
Коалгоритм	6.4	– упорядоченный	2.4.1
Кодирование		– эвристический	2.4.2
– адаптивное	2.2	Мем	5.7
– бинарное	2.2	Мемеплекс	4.7.1
– векторное	2.2	Метаоптимизация	
– вещественное	2.2	– параметрическая	7.1.1
– двоичное	2.2	– динамическая	7.1.2, 7.3.2
– логарифмическое	2.2	– адаптивная	7.3.2
– символьное	2.2	– неадаптивная	7.3.2
– числовое	2.2	– статическая	7.1.1, 7.3.1
Козволюция		– структурная	7.1.3, 7.7
– двухпопуляционная	6.4.1	– динамическая	7.1.3
– многопопуляционная	6.4.1	– адаптивная	7.1.3
– типа соперничества	6.4.1	– неадаптивная	7.1.3
– типа сотрудничества	6.4.1	– статическая	7.1.3
Критерий		– структурно-параметри-ческая	7.1.4, 7.7
– бинарный	8.4	– статическая	7.1.4, 7.7
– оптимальности	1.1.1	– динамическая	7.1.4, 7.7
– унимодальный	1.1.1	– адаптивная	7.1.4
– скользящего допуска	1.3.1	– неадаптивная	7.1.4

Метод		– параллелизма	
– лечения локальным поиском	1.3.2	– глобальная	9.4.1
– настройки параметров	7.2.1	– асинхронная	9.4.1
– однократный	7.2.1, 7.3	– синхронная	9.4.1
– перманентный	7.2.1, 7.4	– диффузная	9.4.3
– нишевания		– комбинированная	9.4.4
– локального	8.5.2	– коэволюционная	9.4.4
– ограничения скрещивания	8.5.2	– островная	9.4.2
– отбора		– асинхронная	9.4.2
– вытеснением	2.5.1	– синхронная	9.4.2
– на основе элитизма	2.5.1	– пространственно вложенная	9.4.4
– отсечением	2.5.1	– расщеплением	9.4.4
– пропорциональный	2.5.1	– типа «хозяин–раб»	9.4.1
– ранговый	2.5.1	Монохромосома	2.2
– селективный	2.5.2	– модифицированная	2.2
– с остатком	2.5.1	Мультихромосома	2.2
– стохастический универсальный	2.5.1	– гомоморфная	2.2
– турнирный	2.5.1	– негомоморфная	2.2
– панмиксии	2.5.2	Мутация	
– последовательного нишевания	2.6	– знаковая	5.2
– разделяющей (сепарирующей)		– переупорядочивающая	5.2
функции	2.6	Ниша	8.5.1
– редукции недопустимых особей		Нишевание	2.6, 5.4.3, 8.5.1
к допустимым	2.6	Оператор	
– рулетки	2.5.1	– адаптивный	2.1
– сетевого оператора	7.3.2	– арифметического сдвига	
– скользящего допуска	1.3.1	– вещественный	2.3.2
– управления параметрами	7.2.1	– ветвления	4.7.4
– адаптивный	7.2.1, 7.5	– вставки	2.4.1
– коэволюционный		– генетический	2.1
– гетерогенный	7.5.2	– геометрического сдвига	
– гомогенный	7.5.1	– вещественный	2.3.2
– детерминированный	7.2.1	– кормления	4.7.3
– самоадаптивный	7.2.1, 7.6	– макромутации	2.3.1
– штрафов		– миграции	2.11
– динамических	2.6	– мутации	
– смертельных	1.3.2, 2.6	– бинарный	2.3.1
– статических	2.6	– гауссовый	2.3.2, 5.6.3
Миграция		– граничный	2.3.2, 5.6.3
– перемещением	9.4.2	– динамический	2.3.2
– репликацией	9.4.2	– зеркальный	2.3.1
Микроалгоритм		– неравномерный	
– генетический	2.7	– генный	2.3.1
Модель		– Михалевича	2.3.2, 5.6.3
– коэволюции		– обменный	2.3.1
– простая	6.4.1	– равномерный	
– композитная	6.4.1	– генный	2.3.1
– некоммутирующей мульти-		– хромосомный	2.3.1
популяции	9.4.4	– случайный	2.3.2
– обучаемой эволюции	2.7	– степенной	2.3.2
– островная	6.4.1	– BGA	2.3.2

– плавления	4.7.3	– дробно-линейное	1.1.2
– прививки	4.7.4	– квадратичное	1.1.2
– разбиения	4.7.3	– линейное	1.1.2
– репликации	2.10	– нелинейное	1.1.2
– сегрегации	2.4.1	– сепарабельное	1.1.2
– скрещивания	4.7.4	– целочисленное	1.1.2
– транслокации бинарный	2.6	– эволюционное	2.10
– удаления	2.4.1	Пространство	
– ускоряющий	2.11	– убеждений	5.6
– транспозиции	2.3.1	– решений	5.9.2
Операция		– целевое	8.1.1
– локальных состязаний	5.4.1	Процедура	
– диссимилиации	5.4.1	– локального поиска	1.4.2
– модифицированная	5.4.2	– меметической эволюции	4.7.1
– конкретизации	5.6.2	– модуляции	3.2.4
– кооперации	5.4.2	– разбиения единичного	
– обобщения	5.6.2	гиперкуба	1.4.1
– разделения	5.6.2	– разбиения единичного отрезка	1.4.1
– распространения	5.6.2	– тасования	4.7.1
– обратного	5.6.2	– улучшения решения	5.9.1
– слияния	5.6.2	Решение	
– случайного выбора	5.1.1	– инициализирующее	5.9.2
Оптимизация		– многообещающее	1.4.3
– альтернативная	6.3.2	– опорное	5.9.1
– безусловная	1.1.2	– целевое	5.9.2
– без ограничений	1.1.2	– эталонное	8.4
– глобальная	1.1.2	Сеть	
– детерминированная	1.1.1	– иммунная	4.1.2
– динамическая	3.2.5	– ограничений	5.6.2
– локальная	1.1.2	Сжатие	
– многоэкстремальная	1.1.2	– клональное	4.1.2
– многоцелевая	8.1	– сетевое	4.1.2
– одномерная	1.1.2	Система	
– однопараметрическая	1.1.2	– асинхронная	9.1.1
– одноэкстремальная	1.1.2	– гетерогенная	9.1.1
– с ограничениями	1.1.2	– гомогенная	9.1.1
– типа неравенств	1.1.2	– искусственных иммунных	
– типа равенств	1.1.2	(<i>AIS</i>)	4.1
– общего вида	1.1.2	– неоднородная	9.1.1
– условная	1.1.2	– однородная	9.1.1
Отображение		– сильносвязанная	9.1.1
– динамическое	9.3	– синхронная	9.1.1
– логистическое	1.4.2	– слабосвязанная	9.1.1
– оптимальное	9.3	– с памятью	
– палаточное	1.4.2	– гибридной	9.1.1
– статическое	9.3	– общей	9.1.1
Принцип Эджворта – Парето	8.1.1	– разделяемой	9.1.1
Программирование		– распределенной	9.1.1
– выпуклое	1.1.2	– <i>MIMD</i>	9.1.1
– геометрическое	1.1.2	– <i>MISD</i>	9.1.1
– дискретное	1.1.2	– <i>SIMD</i>	9.1.1

Стратегия		Участок	
– алгоритма	7	– многообещающий	4.2.3
– выбора		– отобранный	3.3.2
– декомпозиционного	5.7.2	– перспективный	3.3.2
– простого случайного	5.7.2	– элитный	3.3.2
– прямого	5.7.2	Фаза	
– ранжированного	5.7.2	– отталкивания	3.1
– рулеточного	5.7.2	– положительного конфликта	3.1.2
– с запретами	5.7.2	– притяжения	3.1
– динамическая	7.1.2	– промежуточная	3.1.2
– апостериорная (неадаптивная)	7.1.2	Фенотип	2.1
– априорная (адаптивная)	7.1.2	Фронт Парето	8.1.1
– жадная рандомизированная	5.9.2	Функция	
– миграции	9.4.2	– аспирации	1.4.3
– спуска		– влияния	5.6.1
– случайного	5.7.2	– временная	3.2.5
– со случайной перестановкой	5.7.2	– выбора	3.2.5
– статическая	7.1.1	– выпуклая	1.1.1
– структурная	7.1.3	– динамическая	8.3.2
– структурно–параметрическая	7.1.4	– Лагранжа	2.12
– управления		– многоэкстремальная	1.1.1
– интегративная	6.1.3	– мультимодальная	1.1.1
– кооперативная	6.1.3	– овражная	1.1.1
– эволюционная	2.9	– отбора	2.5.2
Субпопуляция	7.5	– подобия	8.5.1
Суброй	7.5.1	– позиномиальная	1.1.1
Суперпопуляция	7.5	– предпочтений	8.4.2
Суперрой	7.5.1	– согласованная	8.4.2
Схема взаимодействия субпопуляций		– строго согласованная	8.4.2
– параллельная	6.4.1	– принятия	5.6.1
– последовательная	6.4.1	– разделяющая	8.5.1
Теорема		– сепарабельная	1.1.1
– шим	2.8	– сравнения	8.4.2
– модифицированная	2.8	– строго выпуклая	1.1.1
– Куна – Такера	2.12	– структурная	3.2.5
Топология соседства		– целевая	1.1.1
– глобально оптимальная	3.1.3	– динамическая	3.2.5
– двумерный тор	3.1.3	– частная	8.1.1
– динамическая	3.1.4	– штрафа	1.3.1
– кластерная	3.1.3	Частица	
– клика	3.1.3	– неправильная	6.4.2
– кольцо	3.1.3	– правильная	6.4.2
– локально оптимальная	3.1.3	Шима	2.8
– статическая	3.1.4	– определяющая длина	2.8
– фон Неймана	3.1.3	– порядок	2.8
Точка		– приспособленность	2.8
– допустимая	1.3.1	Эвристика	
– почти допустимая	1.3.1	– полужадная	1.4.2
– недопустимая	1.3.1	– случайная жадная	1.4.2
– феромонная	3.2.1	Экзон	7.6.1
– центральная	8.2.2	Элитизм	8.3.2

<i>ABC, Artificial Bee Colony</i>	3.3.3	<i>CMEC, Chaotic MEC</i>	5.4.4
<i>ABFO, Adaptive Bacterial Foraging Optimization</i>	4.2.3	<i>COAC, Continuous Orthogonal Ant Colony</i>	3.2.4
<i>ACO, Ant Colony Optimization</i>	3.2	<i>CS, Cuckoo Search</i>	4.5
<i>ACO_R, extended Ant Colony Optimization to continuous domains</i>	3.2	<i>CSS, Charged System Search</i>	5.3.2
<i>ADD, Approximate Descent Direction</i>	1.2.1	<i>CUDA, Compute Unified Device Architecture</i>	9.1.2
<i>AIS, Artificial Immune Systems</i>	4.1	<i>DACO, Direct Application of Ant Colony Optimization</i>	3.2
<i>Altering objective functions algorithm</i>	8.3.2	<i>DFSS, Density FSS</i>	4.7.3
<i>AMRS, Adaptive Maxing Random Search</i>	1.2.2	<i>DHCIAC, Dynamic Hybrid Continuous Interacting Ant Colony</i>	3.2.5
<i>ARPSO, Attractive and Repulsive PSO</i>	3.1.2	<i>DHCIACfind</i>	3.2.5
<i>ARSID, Adaptive Random Search with Intensification and Diversification</i>	1.2.1	<i>DHCIACtrack</i>	3.2.5
<i>AS, Ant System</i>	3.2	<i>DHS, Differential HS algorithm</i>	5.1.2
<i>ASSRS, Adaptive Step Size Random Search</i>	1.2.1	<i>DSPSO, Dynamic Sociometry PSO</i>	3.1.4
<i>AWS, Adaptive Weighted Sum algorithm</i>	8.2.2	<i>Dynamic neighborhood Algorithm</i>	8.3.2
<i>BA, B-algorithm, Bees algorithm</i>	3.3.2	<i>EM, ElectroMagnetism-like algorithm</i>	5.3.1
<i>BAS, Binary Ant System</i>	3.2	<i>EMEC, Extended Mind Evolutionary Computation</i>	5.4.2
<i>BFO, Bacterial Foraging Optimization</i>	4.2.2	<i>F-algorithm, Firefly algorithm</i>	4.3
<i>BFO-GA</i>	4.2.5	<i>FDR, Fitness-Distance Ratio based PSO</i>	3.1.2
<i>BFO-PSO</i>	4.2.5	<i>FFGA, Fonseca and Fleming's multi-objective Genetic Algorithm</i>	8.3.3
<i>BI, Bat-Inspired algorithm</i>	4.7.2	<i>FIPS, Fully Informed Particle Swarm</i>	3.1.2
<i>C-algorithm, Cultural algorithm</i>	5.6	<i>FO, Fly Optimization</i>	4.7
<i>CACO, Continuous Ant Colony Optimization</i>	3.2.2	<i>FSS, Fish School Search</i>	4.7.3
<i>CACS, Continuous Ant Colony System</i>	3.2	<i>FSSRS, Fixed Step Size Random Search</i>	1.2.1
<i>CBFO, Cooperative BFO algorithm</i>	4.2.3	<i>Genitor</i>	2.7
<i>CBFO-H, Cooperative BFO algorithm Hybrid</i>	4.2.3	<i>GCA, Genetic Coevolutionary Algorithm</i>	2.12
<i>CBFO-S, Cooperative BFO algorithm Sequence</i>	4.2.3	<i>GENOCOP, GENetic algorithm for Numerical Optimization for CONstrained Problem</i>	5.6.3
<i>CBPSO, Clubs-Based PSO</i>	3.1.4	<i>GHS, Global HS</i>	5.1.2
<i>CFO, Central Force Optimization</i>	5.2	<i>GIPSO, GPU Island PSO</i>	8.4.1
<i>C-GRASP</i>	1.4.2	<i>GPU, Graphics Processing Unit</i>	9.1.2
<i>CIAC, Continuous Interaction Ant Colony</i>	3.2.3	<i>GRASP, Greedy Randomized Adaptive Search Procedure</i>	1.4.2
<i>CHC, Cross-population selection, Heterogenous recombination and Cataclysmic mutation</i>	2.7	<i>GS, Gravitational Search algorithm</i>	5.2
<i>CLONALG, Clonal Selection Algorithm</i>	4.1.3	<i>GSO, Glowworm Swarm Optimization</i>	4.3.2
<i>CLS, Chaotic Local Search</i>	1.4.2	<i>HCIAC, Hybrid Continuous Interacting Ant Colony</i>	3.2.5
		<i>HMCR, Harmony Memory Considering Rate</i>	5.1.1

<i>HPS, Heuristic Pattern Search</i>	1.2.1	<i>RVNS, Random VNS</i>	1.4.2
<i>HS, Harmony Search algorithm</i>	5.1	<i>SA, Simulated Annealing</i>	1.4.2
<i>IBA, IB algorithm, Improved Bee algorithm</i>	3.3.2	<i>SDPS, Single-Dimension Perturbation Search</i>	1.2.1
<i>ICS, Improved Cuckoo Search</i>	4.5	<i>SDS, Stochastic Diffusion Search</i>	5.5
<i>IHS, Improved HS</i>	5.1.2	<i>SFL, Shuffled Frog-Leaping</i>	4.7.1
<i>ILS, Iterated Local Search</i>	1.4.2	<i>Sigma algorithm</i>	8.3.4
<i>IMEC, Improved MEC</i>	5.4.3	<i>SMP, Symmetric Multi-Processing</i>	9.1.1
<i>IWD, Intelligent Water Drop algorithm</i>	5	<i>SMEC, Simple MEC, Simple Mind Evolutionary Computation</i>	5.4.1
<i>IWO, Invasive Weed Optimization</i>	4.4	<i>SOM, Self Organizing Migrating algorithm</i>	5.8
<i>Lévy flights</i>	1.4.2	<i>SPEA, Strength Pareto Evolutionary Algorithm</i>	8.3.3
<i>LUS, Local Unimodal Sampling</i>	1.2.1	<i>SPEA 2, Strength Pareto Evolutionary Algorithm 2</i>	8.3.3
<i>M-algorithms, Memetic algorithms</i>	5.7	<i>SPSA, Simultaneous Perturbation Stochastic Approximation</i>	1.2.1
<i>MCS, Modified Cuckoo Search</i>	4.5	<i>SS, Scatter Search algorithm</i>	5.9.1
<i>MEC, Mind Evolutionary Computation</i>	5.4	<i>SSG, Saplings Sowing and Growing up</i>	4.7.4
<i>Micro-AIS, Micro Artificial Immune System</i>	4.1.3	<i>SWGR, Sum of Weighted Global Ratios</i>	8.3.3
<i>MOEA/D, Multi-Objective Evolutionary Algorithm based on Decomposition</i>	8.1.2	<i>SWO, Sum of Weighted Objectives</i>	8.2.2
<i>MOGA, Multi-Objective Genetic Algorithm</i>	8.3.3	<i>SWR, Sum of Weighted Ratios</i>	8.3.3
<i>Monkey algorithm</i>	4.6.2	<i>TS, TS algorithm, Tabu Search algorithm</i>	1.4.3
<i>MOPSO, Multi-Objective Particle Swarm Optimization</i>	8.6.2, 9.5.2	<i>UMA, Unified Memory Access</i>	9.1.1
<i>MPI, Message Passing Interface</i>	9.1.2	<i>VEGA, Vector Evaluated Genetic Algorithm</i>	8.3.2
<i>MPP, Massive Parallel Processing</i>	9.1.1	<i>VNS algorithm, Variable Neighborhood Search algorithm</i>	1.4.2
<i>MS, Monkey Search</i>	4.6.1	<i>VPSO, Volitive PSO</i>	4.7.3
<i>MS MOPSO, Multi-Swarm MOPSO</i>	8.6.2	<i>WAR, Weighted Average Ranking</i>	8.3.3
<i>NDS, Non-Dominated Sorting</i>	8.3.3	<i>WFIPS, Weighted FIPS</i>	3.1.2
<i>NPGA, Niche-Pareto Genetic Algorithm</i>	8.3.3	<i>WMR, Weighted Maximum Ranking</i>	8.3.3
<i>NSGA, Non-dominated Sorting Genetic Algorithm</i>	8.3.3		
<i>NSGA II, Non-dominated Sorting Genetic Algorithm II</i>	8.3.3		
<i>NUMA, Non-Uniform Memory Access</i>	9.1.1		
<i>OpenCL, Open Computing Language</i>	9.1.2		
<i>OpenMP, Open Multi-Processing</i>	9.1.2		
<i>PAES, Pareto Achieved Evolution Strategy</i>	8.3.3		
<i>PR, Path Relinking algorithm</i>	5.9.2		
<i>Predator-prey algorithm</i>	8.3.2		
<i>PSO, Particle Swarm Optimization</i>	3.1		
<i>PSO with stereotyping</i>	3.1.4		
<i>RFD, River Formation Dynamics</i>	5		
<i>RFIPS, Ranked FIPS</i>	3.1.2		
<i>RGA, Real-coded Genetic Algorithm</i>	2.2		
<i>RLS, Random Line Search</i>	1.2.1		
<i>RS, Random Search</i>	1.2.1		
<i>RTS, Restricted Tournament Selection</i>	8.5.2		

Оглавление

Предисловие.....	3
Основные обозначения.....	5
Введение.....	8
Глава 1. Постановка задачи поисковой оптимизации и непопуляционные стохастические алгоритмы ее решения.....	14
1.1. Постановка и классификация алгоритмов решения детерминированной задачи поисковой оптимизации.....	14
1.1.1. Постановка задачи.....	14
1.1.2. Классификация задач оптимизации.....	16
1.1.3. Классификация алгоритмов оптимизации.....	19
1.2. Локальная безусловная оптимизация.....	22
1.2.1. Одношаговые алгоритмы.....	22
1.2.2. Многошаговые алгоритмы.....	26
1.2.3. Многоточечные алгоритмы.....	29
1.3. Локальная условная оптимизация.....	33
1.3.1. Алгоритмы на основе сведения задачи условной оптимизации к задаче безусловной оптимизации.....	34
1.3.2. Алгоритмы, не использующие редукцию задачи условной оптимизации к задаче безусловной оптимизации.....	42
1.4. Глобальная оптимизация.....	46
1.4.1. Алгоритмы одномерной редукции.....	46
1.4.2. Алгоритмы случайного поиска.....	53
1.4.3. Поиск с запретами.....	62
<i>Вопросы для самопроверки.....</i>	<i>63</i>
Глава 2. Эволюционные алгоритмы.....	65
2.1. Биологические предпосылки и общая схема эволюционных алгоритмов.....	66
2.2. Кодирование особей.....	69
2.3. Операторы мутации.....	75
2.3.1. Бинарные мутаторы.....	75
2.3.2. Вещественные мутаторы.....	77
2.4. Операторы скрещивания (кроссоверы).....	80
2.4.1. Бинарные кроссоверы.....	80
2.4.2. Вещественные кроссоверы.....	84
2.5. Операторы отбора.....	88
2.5.1. Операторы управления популяцией.....	88
2.5.2. Операторы селекции.....	92
2.6. Другие операторы и процедуры.....	96
2.7. Типовые генетические алгоритмы.....	102
2.8. Теория шим.....	106
2.9. Эволюционная стратегия.....	109
2.10. Эволюционное программирование.....	112
2.11. Дифференциальная эволюция.....	114
2.12. Генетический коэволюционный алгоритм.....	117
2.13. Пример применения генетического алгоритма.....	121
<i>Вопросы для самопроверки.....</i>	<i>125</i>

Глава 3. Алгоритмы роя частиц, колонии муравьев и пчелиного роя	127
3.1. Оптимизация роем частиц.....	127
3.1.1. Канонический алгоритм роя частиц.....	127
3.1.2. Модификации канонического алгоритма роя частиц.....	133
3.1.3. Топологии соседства частиц.....	137
3.1.4. Алгоритмы с динамической топологией соседства частиц.....	141
3.1.5. Гибридный алгоритм на основе роя частиц и имитации отжига.....	143
3.1.6. Пример решения задачи с использованием алгоритма роя частиц.....	146
3.2. Муравьиная оптимизация.....	147
3.2.1. Бионические предпосылки.....	148
3.2.2. Алгоритм непрерывной оптимизации колонией муравьев.....	150
3.2.3. Алгоритм непрерывно взаимодействующей колонии муравьев.....	153
3.2.4. Непрерывный ортогональный алгоритм муравьиной колонии.....	159
3.2.5. Гибридный алгоритм непрерывно взаимодействующей муравьиной колонии.....	163
3.2.6. Пример применения модифицированного алгоритма <i>CIAC</i>	170
3.3. Оптимизация пчелиным роем.....	175
3.3.1. Бионические предпосылки.....	175
3.3.2. Пчелиный алгоритм.....	176
3.3.3. Алгоритм колонии искусственных пчел.....	183
3.3.4. Гибридный алгоритм.....	190
<i>Вопросы для самопроверки</i>	193
Глава 4. Другие популяционные алгоритмы, вдохновленные живой природой	195
4.1. Искусственные иммунные системы.....	195
4.1.1. Биологические основы.....	195
4.1.2. Оптимизация с помощью модели иммунной сети.....	198
4.1.3. Алгоритм на основе искусственной микроиммунной системы.....	200
4.1.4. Пример алгоритма искусственной иммунной сети.....	201
4.2. Бактериальная оптимизация.....	204
4.2.1. Бионические предпосылки.....	204
4.2.2. Канонический алгоритм бактериальной оптимизации.....	205
4.2.3. Кооперативная бактериальная оптимизация.....	210
4.2.4. Алгоритм, использующий эффект роевания бактерий.....	211
4.2.5. Гибридные бактериальные алгоритмы.....	212
4.3. Алгоритмы, вдохновленные роем светлячков.....	213
4.3.1. Алгоритм светлячков.....	213
4.3.2. Алгоритм оптимизации роем светлячков.....	216
4.4. Сорняковый алгоритм.....	218
4.4.1. Биологические основы.....	218
4.4.2. Схема алгоритма.....	219
4.5. Кукушкин поиск.....	221
4.5.1. Биологические предпосылки.....	221
4.5.2. Схема алгоритма.....	222
4.5.3. Эффективность алгоритма.....	223
4.6. Алгоритмы, вдохновленные поведением обезьян.....	227
4.6.1. Обезьяний поиск.....	227
4.6.2. Обезьяний алгоритм.....	228
4.7. Прочие алгоритмы.....	230

4.7.1 Тасующий алгоритм прыгающих лягушек.....	230
4.7.2. Алгоритм, инспирированный летучими мышами.....	231
4.7.3. Поиск косяком рыб.....	234
4.7.4. Алгоритм растущих деревьев.....	237
<i>Вопросы для самопроверки</i>	238
Глава 5. Популяционные алгоритмы, инспирированные неживой природой, человеческим обществом, и другие популяционные алгоритмы.....	239
5.1. Гармонический поиск.....	239
5.1.1. Канонический алгоритм.....	240
5.1.2. Некоторые модификации алгоритма гармонического поиска.....	242
5.1.3. Пример применения модифицированного алгоритма гармонического поиска.....	243
5.2. Алгоритм гравитационного поиска.....	246
5.3. Электромагнитный поиск.....	250
5.3.1. Электромагнитный алгоритм.....	250
5.3.2. Алгоритм поиска системой зарядов.....	254
5.4. Алгоритм эволюции разума.....	255
5.4.1. Простой алгоритм эволюции разума.....	256
5.4.2. Расширенный алгоритм.....	257
5.4.3. Улучшенный алгоритм.....	259
5.4.4. Хаотический алгоритм эволюции разума.....	260
5.5. Стохастический диффузионный поиск.....	261
5.6. Культурный алгоритм.....	263
5.6.1. Общая схема алгоритма.....	263
5.6.2. Построение и коррекция пространства убеждений.....	265
5.6.3. Культурная модификация генетических операторов.....	267
5.7. Меметические алгоритмы.....	268
5.7.1 Общая схема алгоритма.....	268
5.7.2. Гиперэвристические мультимемные адаптивные алгоритмы.....	269
5.7.3. Самоадаптирующиеся мультимемные алгоритмы.....	271
5.8. Самоорганизующийся миграционный алгоритм.....	272
5.9. Алгоритмы рассеянного поиска и прокладки путей.....	274
5.9.1. Алгоритм рассеянного поиска.....	274
5.9.2. Алгоритм прокладки путей.....	278
<i>Вопросы для самопроверки</i>	279
Глава 6. Гибридизация популяционных алгоритмов.....	280
6.1. Общие принципы гибридизации.....	280
6.1.1. Одноуровневая классификация Ванга.....	280
6.1.2. Двухуровневая классификация Эль-Абда и Камэла.....	282
6.1.3. Четырехуровневая классификация Рейдла.....	283
6.2. Вложенные алгоритмы.....	285
6.2.1. Высокоуровневая гибридизация вложением.....	285
6.2.2. Низкоуровневая гибридизация вложением.....	287
6.3. Гибридизация по схеме препроцессор / постпроцессор.....	288
6.3.1. Последовательная гибридизация.....	288
6.3.2. Конвейерная гибридизация.....	290
6.4. Коалгоритмы.....	296
6.4.1. Классификация коэволюционных алгоритмов.....	296
6.4.2. Пример коалгоритма.....	299
<i>Вопросы для самопроверки</i>	301

Глава 7. Метаоптимизация популяционных алгоритмов	302
7.1. Постановка метазадачи оптимизации	302
7.1.1. Статическая параметрическая метаоптимизация	302
7.1.2. Динамическая параметрическая метаоптимизация	304
7.1.3. Структурная метаоптимизация	307
7.1.4. Структурно-параметрическая метаоптимизация	308
7.2. Классификация методов метаоптимизации	309
7.2.1. Классификация Эйбена	309
7.2.2. Классификация Смита – Фогарти	312
7.3. Однократная настройка параметров	313
7.3.1. Статическая параметрическая метаоптимизация	313
7.3.2. Динамическая параметрическая метаоптимизация	315
7.4. Перманентная настройка параметров	316
7.4.1. Схема подхода	316
7.4.2. Задачи кластеризации и собственно метаоптимизации	317
7.4.3. Особенности программной реализации подхода	318
7.4.4. Исследование эффективности подхода	320
7.5. Адаптивное управление параметрами	324
7.5.1. Гомогенные методы	325
7.5.2. Гетерогенные методы	326
7.6. Самоадаптивное управление параметрами	327
7.6.1. Расширение вектора варьируемых параметров базовой задачи	327
7.6.2. Коэволюция типа соперничества	328
7.7. Структурная метаоптимизация	328
<i>Вопросы для самопроверки</i>	330
Глава 8. Популяционные алгоритмы многоцелевой оптимизации	332
8.1. Задача многоцелевой оптимизации (МЦО-задача) и алгоритмы ее решения	332
8.1.1. Постановка задачи	333
8.1.2. Классификация алгоритмов решения МЦО-задачи	336
8.2. Непопуляционные алгоритмы Парето-аппроксимации	340
8.2.1. Сеточные алгоритмы	340
8.2.2. Алгоритмы на основе свертки целевых функций	341
8.3. Популяционные алгоритмы Парето-аппроксимации	345
8.3.1. Лексикографическая турнирная селекция	345
8.3.2. Алгоритмы чередующихся целевых функций	346
8.3.3. Алгоритмы на основе ранжирования агентов	349
8.3.4. Алгоритмы, не использующие ранжирование агентов	353
8.4. Критерии оценки качества Парето-аппроксимации	357
8.4.1. Унарные критерии	358
8.4.2. Бинарные критерии	360
8.5. Методы обеспечения качества Парето-аппроксимации	365
8.5.1. Нишевание	366
8.5.2. Другие методы	367
8.6. Примеры Парето-аппроксимации	370
8.6.1. Парето-аппроксимация на основе генетического алгоритма	370
8.6.2. Аппроксимация с помощью алгоритма роя частиц	373
8.6.3. Аппроксимация на основе алгоритма колонии муравьев	379
8.6.4. Аппроксимация на основе пчелиного алгоритма	380
<i>Вопросы для самопроверки</i>	383

Глава 9. Параллельные популяционные алгоритмы поисковой оптимизации.....	384
9.1. Классификация и основные типы параллельных ЭВМ.....	384
9.1.1. Классификация параллельных вычислительных систем.....	385
9.1.2. Основные типы параллельных ЭВМ.....	389
9.2. Балансировки загрузки параллельной ЭВМ.....	393
9.2.1. Задача оптимального отображения алгоритма на архитектуру параллельной ЭВМ.....	393
9.2.2. Постановка задачи балансировки загрузки.....	395
9.2.3. Методы статической балансировки загрузки.....	396
9.2.4. Методы динамической балансировки загрузки.....	400
9.2.5. Задача согласования алгоритма с архитектурой параллельной ЭВМ.....	403
9.3. Методы распараллеливания популяционных алгоритмов оптимизации.....	404
9.3.1. Глобальная модель параллелизма.....	405
9.3.2. Островная модель параллелизма.....	407
9.3.3. Диффузная модель параллелизма.....	409
9.3.4. Другие модели параллелизма.....	409
9.4. Примеры параллельного решения задач оптимизации.....	411
9.4.1. Параллельный алгоритм роя частиц <i>GIPSO</i>	411
9.4.2. Параллельный алгоритм Парето-аппроксимации роем частиц.....	417
<i>Вопросы для самопроверки.....</i>	<i>419</i>
Литература.....	420
Приложение А. История разработки популяционных алгоритмов поисковой оптимизации.....	422
Приложение Б. Тестовые функции для одноцелевой задачи глобальной оптимизации.....	424
Приложение В. Тестовые задачи многоцелевой оптимизации.....	432
Предметный указатель.....	434

Учебное издание

Карпенко Анатолий Павлович

**СОВРЕМЕННЫЕ АЛГОРИТМЫ
ПОИСКОВОЙ ОПТИМИЗАЦИИ**

Алгоритмы, вдохновленные
природой

Редактор *Л.Н. Лацёва*

Технический редактор *Э.А. Кулакова*

Художник *А.К. Ездовой*

Корректор *А.К. Еникеева*

Компьютерная графика *Ю.В. Мосягина*

Компьютерная верстка *Ю.В. Мосягина*

В оформлении использованы шрифты Студии Артемия Лебедева.

Оригинал-макет подготовлен в Издательстве МГТУ им. Н.Э. Баумана.

Сертификат соответствия № РОСС RU. АЕ51. Н 16228 от 18.06.2012

Подписано в печать 28.06.2014. Формат 70×100 1/16.

Усл. печ. л. 36,4. Тираж 500 экз. Заказ №

Издательство МГТУ им. Н.Э. Баумана.

105005, Москва, 2-я Бауманская ул., д. 5, стр. 1.

press@bmstu.ru

www.baumanpress.ru

Отпечатано в типографии МГТУ им. Н.Э. Баумана.

105005, Москва, 2-я Бауманская ул., д. 5, стр. 1.

baumanprint@gmail.com

**В Издательстве МГТУ им. Н.Э. Баумана
вышла в свет монография
В. А. Овчинникова
«Графы в задачах анализа и синтеза структур
сложных систем»**



Предложен единый подход к определению таких понятий, как ультраграф, гиперграф, ориентированный и неориентированный граф, и рассмотрено использование аппарата теории графов для разработки моделей структур сложных систем, а также постановка задач их синтеза и способы снижения вычислительной сложности алгоритмов на графах.

Выполнен анализ ряда задач проектирования сложных систем, выявлены их общие признаки и характерные особенности.

Для студентов, обучающихся по специальностям, связанным с информатикой. Может быть полезна преподавателям и аспирантам, а также специалистам, работающим в данной области.

Год издания: 2014

Тип издания: монография

Объем: 424 стр. / 34.45 п.л.

Формат: 70x100/16

ISBN: 978-5-7038-3890-7

Информацию о других новых
книгах можно получить
на сайте Издательства МГТУ им. Н.Э. Баумана
www.baumanpress.ru

По вопросам приобретения обращаться в отдел реализации Издательства:
телефон: 8 499 263-60-45;
факс: 8 499 261-45-97