

# Оглавление

|   |           |
|---|-----------|
| Благодарности.....  | 19        |
| Предисловие от издательства .....                                 | 20        |
| Аббревиатуры.....   | 21        |
| <b>Часть I. Введение в эволюционную оптимизацию .....</b>         | <b>27</b> |
| <b>Глава 1. Введение.....</b>                                     | <b>28</b> |
| Обзор главы .....   | 28        |
| 1.1. Терминология.....  | 29        |
| 1.2. Зачем нужна еще одна книга по эволюционным алгоритмам? ..... | 32        |
| 1.3. Предварительные условия .....                                | 34        |
| 1.4. Домашние задания.....  | 35        |
| 1.5. Обозначения .....  | 35        |
| 1.6. План изложения.....  | 38        |
| 1.7. Учебный курс на основе данной книги .....                    | 39        |
| <b>Глава 2. Оптимизация.....</b>                                  | <b>41</b> |
| Краткий обзор главы.....  | 41        |
| 2.1. Неограниченная оптимизация .....                             | 42        |
| 2.2. Ограниченная оптимизация .....                               | 46        |
| 2.3. Многокритериальная оптимизация.....                          | 48        |
| 2.4. Мультимодальная оптимизация.....                             | 51        |
| 2.5. Комбинаторная оптимизация .....                              | 52        |
| 2.6. Восхождение к вершине холма .....                            | 54        |
| 2.6.1. Смещенные оптимизационные алгоритмы.....                   | 59        |
| 2.6.2. Важность симуляций Монте-Карло .....                       | 60        |
| 2.7. Интеллект.....   | 60        |
| 2.7.1. Приспособляемость.....                                     | 61        |
| 2.7.2. Случайность .....  | 61        |
| 2.7.3. Общение .....  | 62        |
| 2.7.4. Обратная связь.....  | 63        |
| 2.7.5. Разведывание и эксплуатация.....                           | 64        |
| 2.8. Заключение .....   | 65        |
| Задачи.....   | 66        |
| Письменные упражнения.....  | 66        |
| Компьютерные упражнения.....                                      | 68        |

|   |            |
|---|------------|
| <b>Часть II. Классические эволюционные алгоритмы .....</b>            | <b>71</b>  |
| <b>Глава 3. Генетические алгоритмы.....</b>                           | <b>72</b>  |
| Краткий обзор главы.....  | 73         |
| 3.1. История генетики.....  | 74         |
| 3.1.1. Чарльз Дарвин.....   | 74         |
| 3.1.2. Грегор Мендель.....  | 77         |
| 3.2. Генетика .....   | 79         |
| 3.3. История генетических алгоритмов .....                            | 81         |
| 3.4. Простой бинарный генетический алгоритм.....                      | 85         |
| 3.4.1. Генетический алгоритм для проектирования роботов .....         | 85         |
| 3.4.2. Отбор и скрещивание.....                                       | 88         |
| 3.4.3. Мутации .....  | 91         |
| 3.4.4. Краткая формулировка генетического алгоритма .....             | 93         |
| 3.4.5. Регулируемые параметры и примеры генетического алгоритма ..... | 93         |
| 3.5. Простой непрерывный генетический алгоритм.....                   | 100        |
| 3.6. Заключение .....   | 105        |
| Задачи.....   | 106        |
| Письменные упражнения.....  | 106        |
| Компьютерные упражнения.....  | 109        |
| <b>Глава 4. Математические модели генетических алгоритмов.....</b>    | <b>111</b> |
| Краткий обзор главы.....  | 111        |
| 4.1. Теория схем .....  | 112        |
| 4.2. Цепи Маркова.....  | 118        |
| 4.3. Обозначения марковской модели для эволюционных алгоритмов .....  | 124        |
| 4.4. Марковские модели генетических алгоритмов.....                   | 129        |
| 4.4.1. Отбор .....  | 129        |
| 4.4.2. Мутации .....  | 130        |
| 4.4.3. Скрещивание.....   | 132        |
| 4.5. Системно-динамические модели генетических алгоритмов .....       | 137        |
| 4.5.1. Отбор .....  | 138        |
| 4.5.2. Мутации .....  | 140        |
| 4.5.3. Скрещивание.....   | 143        |
| 4.6. Заключение .....   | 149        |
| Задачи.....   | 150        |
| Письменные упражнения.....  | 150        |
| Компьютерные упражнения.....  | 151        |
| <b>Глава 5. Эволюционное программирование .....</b>                   | <b>153</b> |
| Краткий обзор главы.....  | 153        |
| 5.1. Непрерывное эволюционное программирование .....                  | 154        |
| 5.2. Конечно-автоматная оптимизация .....                             | 159        |

|   |            |
|---|------------|
| 5.3. Дискретное эволюционное программирование.....                                | 163        |
| 5.4. Дилемма заключенного.....  | 165        |
| 5.5. Задача искусственного муравья .....  | 171        |
| 5.6. Заключение .....   | 176        |
| Задачи.....   | 177        |
| Письменные упражнения.....  | 177        |
| Компьютерные упражнения.....  | 178        |
| <b>Глава 6. Эволюционные стратегии .....</b>                                      | <b>180</b> |
| Краткий обзор главы.....  | 181        |
| 6.1. Эволюционная стратегия $(1 + 1)$ .....                                       | 181        |
| 6.2. Правило 1/5: деривация .....   | 187        |
| 6.3. Эволюционная стратегия $(\mu + 1)$ .....                                     | 191        |
| 6.4. Эволюционные стратегии $(\mu + \lambda)$ и $(\mu, \lambda)$ .....            | 194        |
| Эволюционная стратегия $(\mu, \kappa, \lambda, \rho)$ .....                       | 198        |
| 6.5. Самоадаптивные эволюционные стратегии .....                                  | 198        |
| 6.6. Заключение .....   | 205        |
| Задачи.....   | 206        |
| Письменные упражнения.....  | 206        |
| Компьютерные упражнения.....  | 207        |
| <b>Глава 7. Генетическое программирование .....</b>                               | <b>209</b> |
| Ранние результаты генетического программирования.....                             | 211        |
| Краткий обзор главы.....  | 212        |
| 7.1. Lisp: язык генетического программирования.....                               | 213        |
| 7.2. Основы генетического программирования.....                                   | 219        |
| 7.2.1. Мера приспособленности.....  | 220        |
| 7.2.2. Критерии останова .....  | 221        |
| 7.2.3. Множество терминальных символов.....                                       | 221        |
| 7.2.4. Множество функций.....   | 223        |
| 7.2.5. Инициализация .....  | 225        |
| 7.2.6. Параметры генетической программы .....                                     | 228        |
| 7.3. Генетическое программирование: управление объектом за минимальное время..... | 233        |
| 7.4. Раздувание генетической программы .....                                      | 240        |
| 7.5. Эволюционирующие сущности помимо компьютерных программ.....                  | 243        |
| 7.6. Математический анализ генетического программирования .....                   | 246        |
| 7.6.1. Определения и обозначения.....   | 247        |
| 7.6.2. Отбор и скрещивание .....  | 248        |
| 7.6.3. Мутация и конечные результаты .....  | 253        |
| 7.7. Заключение .....   | 255        |
| Задачи.....   | 258        |

|  |            |
|--|------------|
| Письменные упражнения.....   | 258        |
| Компьютерные упражнения.....   | 259        |
| <b>Глава 8. Вариации эволюционных алгоритмов .....</b>   | <b>263</b> |
| Краткий обзор главы.....   | 263        |
| 8.1. Инициализация.....  | 264        |
| 8.2. Критерии сходимости .....   | 266        |
| 8.3. Представление задачи с помощью кода Грея.....   | 269        |
| 8.4. Элитарность.....  | 274        |
| 8.5. Стационарные и поколенческие алгоритмы.....   | 278        |
| 8.6. Популяционное многообразие.....   | 279        |
| 8.6.1. Дублирующие особи.....  | 280        |
| 8.6.2. Нишевая и видовая рекомбинация.....   | 281        |
| 8.6.3. Ниширование.....  | 283        |
| 8.7. Варианты отбора .....   | 289        |
| 8.7.1. Стохастическая универсальная выборка.....   | 290        |
| 8.7.2. Избыточный отбор.....   | 292        |
| 8.7.3. Сигма-шкалирование .....  | 293        |
| 8.7.4. Ранговый отбор.....   | 295        |
| 8.7.5. Линейное ранжирование.....  | 297        |
| 8.7.6. Турнирный отбор.....  | 300        |
| 8.7.7. Племенные эволюционные алгоритмы .....  | 301        |
| 8.8. Рекомбинация.....   | 303        |
| 8.8.1. Одноточечное скрещивание (в бинарных или непрерывных эволюционных алгоритмах).....            | 304        |
| 8.8.2. Многоточечное скрещивание (в бинарных или непрерывных эволюционных алгоритмах).....           | 304        |
| 8.8.3. Сегментированное скрещивание (в бинарных или непрерывных эволюционных алгоритмах).....        | 304        |
| 8.8.4. Равномерное скрещивание (в бинарных или непрерывных эволюционных алгоритмах).....             | 305        |
| 8.8.5. Многородительское скрещивание (в бинарных или непрерывных эволюционных алгоритмах).....       | 306        |
| 8.8.6. Глобальное однородное скрещивание (в бинарных или непрерывных эволюционных алгоритмах).....   | 306        |
| 8.8.7. Перетасованное скрещивание (в бинарных или непрерывных эволюционных алгоритмах).....          | 307        |
| 8.8.8. Плоское скрещивание и арифметическое скрещивание (в непрерывных эволюционных алгоритмах)..... | 307        |
| 8.8.9. Смешанное скрещивание (в непрерывных эволюционных алгоритмах) .....                           | 308        |
| 8.8.10. Линейное скрещивание (в непрерывных эволюционных алгоритмах) .....                           | 309        |
| 8.8.11. Симулированное бинарное скрещивание (в непрерывных эволюционных алгоритмах).....             | 309        |
| 8.8.12. Резюме.....  | 310        |

|  |            |
|--|------------|
| 8.9. Мутация.....  | 310        |
| 8.9.1. Равномерная мутация с центром в $x_i(k)$ .....                  | 310        |
| 8.9.2. Равномерная мутация с центром в середине поисковой области..... | 311        |
| 8.9.3. Гауссова мутация с центром в $x_i(k)$ .....                     | 311        |
| 8.9.4. Гауссова мутация с центром в середине поисковой области .....   | 312        |
| 8.10. Заключение.....  | 312        |
| Задачи.....  | 313        |
| Письменные упражнения.....   | 313        |
| Компьютерные упражнения.....   | 316        |
| <b>Часть III. Более поздние эволюционные алгоритмы .....</b>           | <b>319</b> |
| <b>Глава 9. Симулированное закаливание .....</b>                       | <b>320</b> |
| Краткий обзор главы.....   | 321        |
| 9.1. Закалка в природе.....  | 321        |
| 9.2. Простой алгоритм симулированного закаливания.....                 | 323        |
| 9.3. Режимы охлаждения.....  | 326        |
| 9.3.1. Линейное охлаждение .....                                       | 326        |
| 9.3.2. Экспоненциальное охлаждение.....                                | 327        |
| 9.3.3. Обратное охлаждение.....  | 327        |
| 9.3.4. Логарифмическое охлаждение.....                                 | 330        |
| 9.3.5. Обратное линейное охлаждение.....                               | 332        |
| 9.3.6. Размерно-зависимое охлаждение .....                             | 334        |
| 9.4. Вопросы реализации.....   | 338        |
| 9.4.1. Генерирование кандидатного решения .....                        | 338        |
| 9.4.2. Реинициализация .....   | 338        |
| 9.4.3. Отслеживание лучшего кандидатного решения.....                  | 339        |
| 9.5. Заключение .....  | 339        |
| Задачи.....  | 340        |
| Письменные упражнения.....   | 340        |
| Компьютерные упражнения.....   | 341        |
| <b>Глава 10. Оптимизация на основе муравьиной кучи .....</b>           | <b>343</b> |
| Краткий обзор главы.....   | 346        |
| 10.1. Модели феромона.....   | 346        |
| 10.2. Муравьиная система .....   | 350        |
| 10.3. Непрерывная оптимизация.....                                     | 356        |
| 10.4. Другие муравьиные системы.....                                   | 360        |
| 10.4.1. Минимаксная муравьиная система .....                           | 360        |
| 10.4.2. Система муравьиной кучи .....                                  | 362        |
| 10.4.3. Еще больше муравьиных систем.....                              | 366        |
| 10.5. Теоретические результаты.....                                    | 368        |

|   |            |
|---|------------|
| 10.6. Заключение.....   | 369        |
| Задачи.....   | 371        |
| Письменные упражнения.....  | 371        |
| Компьютерные упражнения.....  | 373        |
| <b>Глава 11. Оптимизация на основе роя частиц.....</b>                        | <b>374</b> |
| Краткий обзор главы.....  | 376        |
| 11.1. Базовый алгоритм оптимизации на основе роя частиц.....                  | 377        |
| Топологии роя частиц.....   | 380        |
| 11.2. Ограничение скорости.....   | 381        |
| 11.3. Коэффициенты инерционного взвешивания и сужения.....                    | 383        |
| 11.3.1. Инерционное взвешивание.....  | 383        |
| 11.3.2. Коэффициент сужения.....  | 384        |
| 11.3.3. Стабильность оптимизации на основе роя частиц.....                    | 387        |
| 11.4. Глобальные обновления скорости.....                                     | 393        |
| 11.5. Полноинформированный рой частиц.....                                    | 396        |
| 11.6. Самообучение на ошибках.....  | 400        |
| 11.7. Заключение.....   | 403        |
| Задачи.....   | 405        |
| Письменные упражнения.....  | 405        |
| Компьютерные упражнения.....  | 407        |
| <b>Глава 12. Дифференциальная эволюция.....</b>                               | <b>409</b> |
| Краткий обзор главы.....  | 410        |
| 12.1. Базовый алгоритм дифференциальной эволюции.....                         | 410        |
| 12.2. Вариации дифференциальной эволюции.....                                 | 413        |
| 12.2.1. Пробные векторы.....  | 413        |
| 12.2.2. Мутантные векторы.....  | 416        |
| 12.2.3. Корректировка коэффициента шкалирования.....                          | 421        |
| 12.3. Дискретная оптимизация.....   | 425        |
| 12.3.1. Смешанно-целочисленная дифференциальная эволюция.....                 | 425        |
| 12.3.2. Дискретная дифференциальная эволюция.....                             | 426        |
| 12.4. Дифференциальная эволюция и генетические алгоритмы.....                 | 428        |
| 12.5. Заключение.....   | 431        |
| Задачи.....   | 431        |
| Письменные упражнения.....  | 431        |
| Компьютерные упражнения.....  | 432        |
| <b>Глава 13. Алгоритмы оценивания вероятностных распределений.....</b>        | <b>434</b> |
| Краткий обзор главы.....  | 435        |
| 13.1. Алгоритмы оценивания вероятностных распределений: основные понятия..... | 436        |
| 13.1.1. Простой алгоритм оценивания вероятностного распределения.....         | 436        |

|  |            |
|--|------------|
| 13.1.2. Вычисление статистических показателей.....   | 437        |
| 13.2. Алгоритмы оценивания вероятностных распределений на основе<br>статистических показателей первого порядка ..... | 438        |
| 13.2.1. Алгоритм одномерного маргинального распределения (UMDA) .....  | 438        |
| 13.2.2. Компактный генетический алгоритм (сGA).....  | 441        |
| 13.2.3. Популяционное инкрементное самообучение (PBIL).....  | 445        |
| 13.3. Алгоритмы оценивания вероятностных распределений на основе<br>статистических показателей второго порядка.....  | 449        |
| 13.3.1. Максимизация взаимной информации для кластеризации входных<br>данных (MIMIC).....                            | 450        |
| 13.3.2. Объединение оптимизаторов с деревьями взаимной информации<br>(COMIT) .....                                   | 456        |
| 13.3.3. Алгоритм двумерного маргинального распределения (BMEDA) .....  | 463        |
| 13.4. Алгоритмы оценивания многомерных вероятностных распределений.....  | 467        |
| 13.4.1. Расширенный компактный генетический алгоритм (ECGA).....   | 467        |
| 13.4.2. Другие алгоритмы оценивания многомерных вероятностных<br>распределений .....                                 | 471        |
| 13.5. Алгоритмы оценивания непрерывных вероятностных распределений.....  | 471        |
| 13.5.1. Алгоритм одномерного маргинального непрерывного распределения.....   | 473        |
| 13.5.2. Непрерывное популяционное инкрементное самообучение.....   | 474        |
| 13.6. Заключение.....  | 478        |
| Задачи.....  | 481        |
| Письменные упражнения.....   | 481        |
| Компьютерные упражнения.....   | 482        |
| <b>Глава 14. Биогеографическая оптимизация.....</b>  | <b>484</b> |
| Краткий обзор главы.....   | 485        |
| 14.1. Биогеография .....   | 485        |
| Математическая модель биогеографии.....  | 488        |
| 14.2. Биогеография как процесс оптимизации.....  | 492        |
| 14.3. Биогеографическая оптимизация.....   | 495        |
| 14.4. Расширения алгоритма биогеографической оптимизации.....  | 500        |
| 14.4.1. Миграционные кривые.....   | 501        |
| 14.4.2. Смешанная миграция .....   | 503        |
| 14.4.3. Другие подходы к биогеографической оптимизации.....  | 505        |
| 14.4.4. Алгоритм биогеографической оптимизации и генетические алгоритмы.....   | 508        |
| 14.5. Заключение.....  | 510        |
| Задачи.....  | 516        |
| Письменные упражнения.....   | 516        |
| Компьютерные упражнения.....   | 517        |
| <b>Глава 15. Культурные алгоритмы.....</b>   | <b>519</b> |
| Краткий обзор главы.....   | 520        |

|   |            |
|---|------------|
| 15.1. Сотрудничество и конкуренция.....                                 | 521        |
| Бар «Эль Фароль».....   | 522        |
| Другие примеры.....   | 523        |
| 15.2. Пространства убеждений в культурных алгоритмах.....               | 525        |
| 15.3. Культурно-эволюционное программирование.....                      | 529        |
| 15.4. Адаптивно-культурная модель.....                                  | 532        |
| 15.5. Заключение.....   | 541        |
| Задачи.....   | 542        |
| Письменные упражнения.....  | 542        |
| Компьютерные упражнения.....  | 543        |
| <b>Глава 16. Опозиционное самообучение.....</b>                         | <b>544</b> |
| Краткий обзор главы.....  | 545        |
| 16.1. Определения и идеи оппозиции.....                                 | 545        |
| 16.1.1. Отраженные противоположности и противоположности по модулю..... | 545        |
| 16.1.2. Частичные противоположности.....                                | 547        |
| 16.1.3. Противоположности 1-го рода и противоположности 2-го рода.....  | 548        |
| 16.1.4. Квазипротивоположности и сверхпротивоположности.....            | 550        |
| 16.2. Алгоритмы оппозиционной эволюции.....                             | 551        |
| 16.3. Опозиционные вероятности.....                                     | 558        |
| 16.4. Коэффициент перескока.....  | 563        |
| 16.5. Опозиционная комбинаторная оптимизация.....                       | 565        |
| 16.6. Дуальное самообучение.....  | 569        |
| 16.7. Заключение.....   | 571        |
| Задачи.....   | 572        |
| Письменные упражнения.....  | 572        |
| Компьютерные упражнения.....  | 573        |
| <b>Глава 17. Другие эволюционные алгоритмы.....</b>                     | <b>574</b> |
| 17.1. Табуированный поиск.....  | 575        |
| 17.2. Алгоритм искусственного косяка рыб.....                           | 576        |
| 17.2.1. Случайное поведение.....  | 577        |
| 17.2.2. Поведение преследования.....                                    | 578        |
| 17.2.3. Роевое поведение.....   | 578        |
| 17.2.4. Поисковое поведение.....  | 579        |
| 17.2.5. Скачущее поведение.....   | 579        |
| 17.2.6. Резюме алгоритма искусственного косяка рыб.....                 | 580        |
| 17.3. Оптимизатор на основе группового поиска.....                      | 581        |
| 17.4. Алгоритм перемешанных лягушачьих прыжков.....                     | 585        |
| 17.5. Светлячковый алгоритм.....  | 587        |
| 17.6. Оптимизация на основе бактериальной кормодобычи.....              | 589        |
| 17.7. Алгоритм искусственной пчелиной семьи.....                        | 593        |



|  |            |
|--|------------|
| 17.8. Алгоритм гравитационного поиска.....                   | 596        |
| 17.9. Поиск гармонии.....                                    | 598        |
| 17.10. Оптимизация по принципу учитель–ученик.....           | 601        |
| 17.11. Заключение.....                                       | 605        |
| Задачи.....  | 607        |
| Письменные упражнения.....                                   | 607        |
| Компьютерные упражнения.....                                 | 608        |
| <b>Часть IV. Специальные типы оптимизационных задач.....</b> | <b>609</b> |
| <b>Глава 18. Комбинаторная оптимизация.....</b>              | <b>610</b> |
| Краткий обзор главы.....                                     | 611        |
| 18.1. Задача коммивояжера.....                               | 612        |
| 18.2. Инициализация задачи коммивояжера.....                 | 614        |
| 18.2.1. Инициализация на основе ближайшего соседа.....       | 614        |
| 18.2.2. Инициализация на основе кратчайшего ребра.....       | 616        |
| 18.2.3. Инициализации на основе вставки.....                 | 618        |
| 18.2.4. Стохастическая инициализация.....                    | 620        |
| 18.3. Представления задачи коммивояжера и скрещивание.....   | 621        |
| 18.3.1. Представление в виде пути.....                       | 621        |
| 18.3.2. Представление в виде смежности.....                  | 626        |
| 18.3.3. Порядковое представление.....                        | 630        |
| 18.3.4. Матричное представление.....                         | 631        |
| 18.4. Мутация в задаче коммивояжера.....                     | 635        |
| 18.4.1. Инверсия.....  | 635        |
| 18.4.2. Вставка.....   | 635        |
| 18.4.3. Перенесение.....                                     | 636        |
| 18.4.4. Взаимообмен.....                                     | 636        |
| 18.5. Эволюционный алгоритм для задачи коммивояжера.....     | 636        |
| 18.6. Задача раскраски графа.....                            | 643        |
| 18.7. Заключение.....  | 648        |
| Задачи.....  | 650        |
| Письменные упражнения.....                                   | 650        |
| Компьютерные упражнения.....                                 | 651        |
| <b>Глава 19. Ограниченная оптимизация.....</b>               | <b>652</b> |
| Краткий обзор главы.....                                     | 653        |
| 19.1. Подходы на основе штрафной функции.....                | 654        |
| 19.1.1. Методы внутренней точки.....                         | 655        |
| 19.1.2. Внешние методы.....                                  | 657        |
| 19.2. Популярные методы обработки ограничений.....           | 660        |
| 19.2.1. Статические штрафные методы.....                     | 660        |

|  |            |
|--|------------|
| 19.2.2. Превосходство допустимых точек .....                       | 660        |
| 19.2.3. Эклектичный эволюционный алгоритм .....                    | 661        |
| 19.2.4. Козволюционные штрафы .....                                | 662        |
| 19.2.5. Динамические штрафные методы .....                         | 664        |
| 19.2.6. Адаптивные штрафные методы.....                            | 667        |
| 19.2.7. Сегрегированный генетический алгоритм .....                | 668        |
| 19.2.8. Самоадаптивная формулировка приспособленности .....        | 668        |
| 19.2.9. Самоадаптивная штрафная функция .....                      | 670        |
| 19.2.10. Адаптивная сегрегационная обработка ограничений .....     | 672        |
| 19.2.11. Поведенческая память .....                                | 673        |
| 19.2.12. Стохастическое ранжирование .....                         | 675        |
| 19.2.13. Нишевой штрафной метод .....                              | 676        |
| 19.3. Специальные представления и специальные операторы .....      | 677        |
| 19.3.1. Специальные представления .....                            | 678        |
| 19.3.2. Специальные операторы .....                                | 681        |
| 19.3.3. Алгоритм Genosop .....                                     | 683        |
| 19.3.4. Алгоритм Genosop II .....                                  | 684        |
| 19.3.5. Алгоритм Genosop III .....                                 | 684        |
| 19.4. Другие подходы к ограниченной оптимизации .....              | 686        |
| 19.4.1. Культурные алгоритмы .....                                 | 687        |
| 19.4.2. Многокритериальная оптимизация .....                       | 687        |
| 19.5. Ранжирование кандидатных решений .....                       | 688        |
| 19.5.1. Ранжирование по максимальному нарушению ограничений .....  | 689        |
| 19.5.2. Ранжирование по порядку следования ограничений .....       | 689        |
| 19.5.3. Метод сравнений $\epsilon$ -уровня .....                   | 690        |
| 19.6. Сравнение методов обработки ограничений .....                | 691        |
| 19.7. Заключение .....   | 695        |
| Задачи .....   | 699        |
| Письменные упражнения .....  | 699        |
| Компьютерные упражнения .....                                      | 701        |
| <b>Глава 20. Многокритериальная оптимизация .....</b>              | <b>703</b> |
| Краткий обзор главы .....  | 705        |
| 20.1. Оптимальность по Парето .....                                | 705        |
| 20.2. Цели многокритериальной оптимизации .....                    | 711        |
| 20.2.1. Гиперобъем .....   | 715        |
| 20.2.2. Относительный охват .....                                  | 718        |
| 20.3. Непаретоориентированные эволюционные алгоритмы .....         | 719        |
| 20.3.1. Методы агрегирования .....                                 | 719        |
| 20.3.2. Генетический алгоритм с векторным оцениванием (VEGA) ..... | 722        |
| 20.3.3. Лексикографическое упорядочение .....                      | 724        |
| 20.3.4. Метод $\epsilon$ -ограничений .....                        | 724        |
| 20.3.5. Гендерные подходы .....                                    | 726        |

|   |            |
|---|------------|
| 20.4. Паретоориентированные эволюционные алгоритмы.....   | 728        |
| 20.4.1. Эволюционные многокритериальные оптимизаторы.....   | 729        |
| 20.4.2. $\epsilon$ -ориентированный многокритериальный эволюционный алгоритм ( $\epsilon$ -MOEA)..... | 731        |
| 20.4.3. Генетический алгоритм с сортировкой по уровню недоминирования (NSGA).....                     | 734        |
| 20.4.4. Многокритериальный генетический алгоритм (MOGA).....  | 737        |
| 20.4.5. Генетический алгоритм с нишированием по Парето (NPGA).....                                    | 739        |
| 20.4.6. Эволюционный алгоритм с расчетом силы Парето (SPEA).....                                      | 740        |
| 20.4.7. Эволюционная стратегия с архивированием по Парето (PAES).....                                 | 749        |
| 20.5. Многокритериальная биогеографическая оптимизация.....   | 750        |
| 20.5.1. Биогеографическая оптимизация с векторным оцениванием.....                                    | 750        |
| 20.5.2. Алгоритм ВВО с сортировкой по уровню недоминирования.....                                     | 751        |
| 20.5.3. Алгоритм ВВО с нишированием по Парето.....  | 752        |
| 20.5.4. Алгоритм ВВО с расчетом силы Парето.....  | 754        |
| 20.5.5. Симуляции многокритериальной биогеографической оптимизации.....                               | 755        |
| 20.6. Заключение.....   | 757        |
| Задачи.....   | 761        |
| Письменные упражнения.....  | 761        |
| Компьютерные упражнения.....  | 763        |
| <b>Глава 21. Дорогостоящие, шумные и динамические функции приспособленности.....</b>                  | <b>765</b> |
| Краткий обзор главы.....  | 766        |
| 21.1. Дорогостоящие функции приспособленности.....  | 767        |
| 21.1.1. Аппроксимирование функции приспособленности.....  | 770        |
| 21.1.2. Аппроксимирование трансформированных функций.....   | 783        |
| 21.1.3. Как применять аппроксимации приспособленности в эволюционных алгоритмах.....                  | 785        |
| 21.1.4. Множественные модели.....   | 789        |
| 21.1.5. Переподгонка.....   | 792        |
| 21.1.6. Оценивание аппроксимационных методов.....   | 793        |
| 21.2. Динамические функции приспособленности.....   | 796        |
| 21.2.1. Предсказательный эволюционный алгоритм.....   | 799        |
| 21.2.2. Иммигрантские схемы.....  | 801        |
| 21.2.3. Подходы на основе памяти.....   | 807        |
| 21.2.4. Оценивание результативности динамической оптимизации.....                                     | 809        |
| 21.3. Шумные функции приспособленности.....   | 810        |
| 21.3.1. Многократное взятие проб.....   | 812        |
| 21.3.2. Оценивание приспособленности.....   | 816        |
| 21.3.3. Эволюционный алгоритм на основе фильтра Калмана.....  | 816        |
| 21.4. Заключение.....   | 819        |
| Задачи.....   | 822        |

|  |            |
|--|------------|
| Письменные упражнения.....   | 822        |
| Компьютерные упражнения.....   | 824        |
| <b>Часть V. Приложения .....</b>   | <b>825</b> |
| <b>Приложение А. Несколько практических советов.....</b>   | <b>826</b> |
| А.1. Проверка на наличие ошибок.....   | 826        |
| А.2. Эволюционные алгоритмы являются стохастическими .....   | 827        |
| А.3. Небольшие изменения могут иметь большой эффект .....  | 828        |
| А.4. Большие изменения могут иметь малый эффект .....  | 828        |
| А.5. Популяции имеют много информации .....  | 829        |
| А.6. Поощрение многообразия.....   | 829        |
| А.7. Использование специфичной на конкретной задаче информации .....                                   | 830        |
| А.8. Сохраняйте свои результаты как можно чаще .....   | 830        |
| А.9. Понимание статистической значимости .....   | 830        |
| А.10. Пишите хорошо.....   | 831        |
| А.11. Акцент на теории.....  | 831        |
| А.12. Акцент на практике.....  | 832        |
| <b>Приложение В. Теорема об отсутствии бесплатных обедов и тестирование<br/>результативности .....</b> | <b>833</b> |
| В.1. Теорема об отсутствии бесплатных обедов .....   | 834        |
| В.2. Тестирование результативности .....   | 844        |
| В.2.1. Преувеличения на основе результатов симуляций.....  | 845        |
| В.2.2. Как отчитываться (и как не отчитываться) о результатах симулирования .....                      | 848        |
| В.2.3. Случайные числа.....  | 856        |
| В.2.4. t-тесты.....  | 859        |
| В.2.5. F-тесты.....  | 866        |
| В.3. Заключение .....  | 872        |
| <b>Приложение С. Эталонные оптимизационные функции.....</b>  | <b>873</b> |
| С.1. Неограниченные эталоны .....  | 874        |
| С.1.1. Сферическая функция .....   | 874        |
| С.1.2. Функция Экли .....  | 875        |
| С.1.3. Тестовая функция Экли .....   | 876        |
| С.1.4. Функция Розенброка.....   | 876        |
| С.1.5. Функция Флетчера .....  | 877        |
| С.1.6. Функция Гриванка .....  | 878        |
| С.1.7. Штрафная функция #1 .....   | 879        |
| С.1.8. Штрафная функция #2.....  | 879        |
| С.1.9. Квартичная функция .....  | 880        |
| С.1.10. Десятистепенная функция.....   | 881        |
| С.1.11. Функция Растригина .....   | 882        |

|  |     |
|--|-----|
| С.1.12. Функция двойной суммы Швевеля.....                               | 883 |
| С.1.13. Функция $\max$ Швевеля.....                                      | 883 |
| С.1.14. Функция Швевеля абсолютной величины .....                        | 884 |
| С.1.15. Синусоидальная функция Швевеля .....                             | 885 |
| С.1.16. Ступенчатая функция.....   | 885 |
| С.1.17. Функция абсолютной величины.....                                 | 886 |
| С.1.18. Функция лисьей норы Шекеля .....                                 | 887 |
| С.1.19. Функция Михалевича.....  | 887 |
| С.1.20. Функция синусоидальной огибающей.....                            | 888 |
| С.1.21. Функция в форме упаковки для яиц.....                            | 889 |
| С.1.22. Функция Вейерштрасса.....  | 889 |
| С.2. Ограниченные эталоны .....  | 890 |
| С.2.1. Функция С01.....  | 891 |
| С.2.2. Функция С02.....  | 891 |
| С.2.3. Функция С03.....  | 892 |
| С.2.4. Функция С04 .....   | 892 |
| С.2.5. Функция С05.....  | 892 |
| С.2.6. Функция С06.....  | 893 |
| С.2.7. Функция С07.....  | 893 |
| С.2.8. Функция С08.....  | 893 |
| С.2.9. Функция С09.....  | 894 |
| С.2.10. Функция С10.....   | 894 |
| С.2.11. Функция С11 .....  | 894 |
| С.2.12. Функция С12 .....  | 895 |
| С.2.13. Функция С13 .....  | 895 |
| С.2.14. Функция С14 .....  | 895 |
| С.2.15. Функция С15 .....  | 896 |
| С.2.16. Функция С16 .....  | 896 |
| С.2.17. Функция С17.....   | 897 |
| С.2.18. Функция С18 .....  | 897 |
| С.2.19. Резюме ограниченных эталонов .....                               | 897 |
| С.3. Многокритериальные эталоны.....                                     | 898 |
| С.3.1. Неограниченная многокритериальная оптимизационная задача 1 .....  | 899 |
| С.3.2. Неограниченная многокритериальная оптимизационная задача 2 .....  | 900 |
| С.3.3. Неограниченная многокритериальная оптимизационная задача 3 .....  | 901 |
| С.3.4. Неограниченная многокритериальная оптимизационная задача 4 .....  | 901 |
| С.3.5. Неограниченная многокритериальная оптимизационная задача 5 .....  | 902 |
| С.3.6. Неограниченная многокритериальная оптимизационная задача 6 .....  | 903 |
| С.3.7. Неограниченная многокритериальная оптимизационная задача 7 .....  | 904 |
| С.3.8. Неограниченная многокритериальная оптимизационная задача 8 .....  | 904 |
| С.3.9. Неограниченная многокритериальная оптимизационная задача 9 .....  | 905 |
| С.3.10. Неограниченная многокритериальная оптимизационная задача 10..... | 906 |
| С.4. Динамические эталоны .....  | 907 |

|  |            |
|--|------------|
| С.4.1. Полное описание динамического эталона .....     | 907        |
| С.4.2. Упрощенное описание динамического эталона ..... | 914        |
| С.5. Шумные эталоны.....                               | 915        |
| С.6. Задачи коммивояжера.....                          | 915        |
| С.7. Устранение смещения в поисковом пространстве..... | 919        |
| С.7.1. Сдвиги.....                                     | 919        |
| С.7.2. Матрицы поворота .....                          | 921        |
| <b>Предметный указатель .....</b>                      | <b>925</b> |

# Часть I



## Введение в эволюционную ОПТИМИЗАЦИЮ

# Глава 1

## Введение

*И подлинно: спроси у скота, и научит тебя, у птицы небесной, и возвестит тебе; или побеседуй с землею, и наставит тебя, и скажут тебе рыбы морские.*

– Иов 12:7-9

В настоящей книге обсуждаются подходы к решению оптимизационных задач. В частности, мы<sup>1</sup> обсуждаем эволюционные алгоритмы (evolutionary algorithms, EA) для оптимизации. Несмотря на то что данная книга содержит немного математической теории, эту книгу не следует рассматривать как пособие по математике. Это скорее пособие по инженерному делу или прикладной информатике. Рассматриваемые в настоящей книге подходы к оптимизации приводятся с целью их потенциальной реализации в программном обеспечении. Задача настоящей книги – представить алгоритмы эволюционной оптимизации самым ясным, но строгим образом, а также предоставить достаточный объем продвинутого материала и ссылок, с тем чтобы читатель был готов внести новый материал в современное состояние дел.

### Обзор главы

Эта глава начинается в разделе 1.1 с краткого обзора математических обозначений, которые мы используем в данной книге. Читателю также, возможно, будет полезен список аббревиатур, начинающихся со

---

<sup>1</sup> В этой книге используется общепринятая практика обращения к третьему лицу с помощью местоимения «мы». Иногда в книге используются местоимения «мы» для ссылок на читателя и автора. В иных случаях в книге используется местоимение «мы», для того чтобы показать, что в ней говорится от имени всего состава преподавателей и исследователей в области эволюционных алгоритмов и оптимизации. Разница должна быть ясна из контекста. Не придавайте слишком большое значение применению местоимения «мы»; это вопрос стиля изложения, а не претензии на авторитет.



страницы 21. В разделе 1.2 приведены некоторые причины, по которым я решил написать эту книгу об эволюционных алгоритмах, чего я надеюсь достичь с ее помощью и почему я думаю, что она отличается от всех других имеющихся превосходных книг по эволюционным алгоритмам. В разделе 1.3 рассматриваются предварительные условия, которым, как мы надеемся, читатели данной книги отвечают. В разделе 1.4 обсуждается общая организация домашних заданий, приводимых в данной книге, и наличие справочника с решениями задач. В разделе 1.5 резюмированы математические обозначения, которые мы используем в этой книге. Читателю рекомендуется регулярно обращаться к этому разделу при появлении незнакомых обозначений, а также самостоятельно применять их в домашних заданиях и в собственных исследованиях. В разделе 1.6 дается подробный план изложения в книге. Данный раздел переходит в раздел 1.7, в котором приводятся важные указания инструктору относительно некоторых приемов преподавания курса на основе этой книги. Данный раздел также дает преподавателю некоторые советы о том, какие главы являются более важными, чем другие.

## 1.1. Терминология

Некоторые авторы для обозначения эволюционных алгоритмов используют термин *эволюционные вычисления*. Этим подчеркивается тот факт, что эволюционные алгоритмы реализуются в компьютерах. Однако эволюционные вычисления могут относиться к алгоритмам, которые не используются для оптимизации; например, первые генетические алгоритмы не использовались для оптимизации как таковой, а были предназначены для изучения процесса естественного отбора (см. главу 3). Данная книга нацелена на алгоритмы эволюционной оптимизации, которые являются более специализированными, чем эволюционные вычисления.

Другие авторы для обозначения эволюционных алгоритмов используют термин *популяционно-ориентированная оптимизация*. Этим подчеркивается тот факт, что эволюционные алгоритмы обычно состоят из популяции кандидатных решений некоторой задачи, и по прошествии времени популяция эволюционирует к более качественному решению задачи. Однако многие эволюционные алгоритмы на каждой итерации могут состоять только из одного кандидатного решения (например, эволюционные стратегии и восхождения к вершине холма). Эволюционные алгоритмы являются более общими, чем популяционно-ориен-

тированная оптимизация, потому что эволюционные алгоритмы включают в себя алгоритмы с одной особью.

Отдельные авторы для обозначения эволюционных алгоритмов используют термин «компьютерный интеллект» или «вычислительный интеллект». Это часто делается для того, чтобы отличить эволюционные алгоритмы от экспертных систем, которые традиционно называются искусственным интеллектом. Экспертные системы моделируют дедуктивные рассуждения, в то время как эволюционные алгоритмы моделируют индуктивные рассуждения. Однако иногда эволюционные алгоритмы рассматриваются как разновидность искусственного интеллекта. Компьютерный интеллект – это более общий термин, чем эволюционный алгоритм, и включает в себя такие технологии, как нейронные сети, нечеткие системы и искусственная жизнь. Эти технологии могут использоваться для приложений, которые не являются оптимизационными. Поэтому, в зависимости от той или иной точки зрения, эволюционные алгоритмы вполне могут быть более общими или более конкретными, чем компьютерный интеллект.

*Мягкие вычисления* – это еще один термин, который связан с эволюционными алгоритмами. Мягкие вычисления противопоставляются жестким вычислениям. Жесткие вычисления относятся к точным, конкретным, численно строгим вычислениям. Под «мягкими» вычислениями понимаются менее точные вычисления, например те, которые люди выполняют в повседневной жизни. Алгоритмы мягких вычислений вычисляют, как правило, хорошие (но неточные) решения сложных, шумных, мультимодальных и многокритериальных задач. Поэтому эволюционные алгоритмы являются подмножеством мягких вычислений.

Другие авторы для обозначения эволюционных алгоритмов используют такие термины, как природообусловленные вычисления или биообусловленные вычисления. Вместе с тем некоторые эволюционные алгоритмы, такие как дифференциальная эволюция и алгоритмы оценивания вероятностных распределений, возможно, не являются природообусловленными. Другие эволюционные алгоритмы, такие как эволюционные стратегии и оппозиционное самообучение, имеют очень слабую связь с естественными процессами. Эволюционные алгоритмы являются более общими, чем природообусловленные алгоритмы, потому что эволюционные алгоритмы включают в себя не биологически обусловленные алгоритмы.

Еще одним часто используемым термином для обозначения эволюционных алгоритмов является машинное (само)обучение. Машинное (само)обучение – это область исследования компьютерных алгоритмов,

которые автоматически обучаются на опыте. Однако эта область часто включает в себя множество алгоритмов, отличных от эволюционных алгоритмов. Машинное (само)обучение обычно считается более широкой областью, чем эволюционные алгоритмы, и включает в себя такие подобласти, как самообучение с подкреплением, нейронные сети, кластеризация, опорно-векторные машины и другие.

Некоторые авторы для обозначения эволюционных алгоритмов предпочитают использовать термин *эвристические алгоритмы*. Термин «эвристика» происходит от греческого слова *εὑρισκω*, которое транслитерируется как «эврика» на русском языке (*eurisko* на английском). Это слово означает «отыскиваю» или «открываю», а также является источником восклицания «эврика», которое мы используем, чтобы выразить радость, когда мы что-то обнаруживаем или решаем задачу. Эвристические алгоритмы – это методы, в которых для решения задачи используются эмпирические правила или подходы на основе здравого смысла. От эвристических алгоритмов, как правило, не рассчитывают получить лучший ответ на поставленную задачу, а ожидают только те решения, которые «достаточно близки» к самому лучшему. Термин *метаэвристика* используется для описания семейства эвристических алгоритмов. Большинство, если не все, эволюционных алгоритмов, которые мы обсуждаем в этой книге, могут быть реализованы по-разному и с целым рядом различных вариаций и параметров. Поэтому их все можно назвать *метаэвристикой*. Например, семейство всех алгоритмов оптимизации на основе муравьиной кучи можно назвать *метаэвристикой муравьиной кучи*.

Большинство авторов проводит различие между эволюционными алгоритмами и роевым интеллектом. Алгоритм роевого интеллекта основан на роях, встречающихся в природе (например, рои муравьев или стаи птиц). Оптимизация на основе муравьиной кучи (глава 10) и оптимизация на основе роя частиц (глава 11) – это два широко известных роевых алгоритма, и многие исследователи настаивают на том, что их не следует классифицировать как эволюционные алгоритмы. Однако некоторые авторы рассматривают роевой интеллект как подмножество эволюционных алгоритмов. Например, один из изобретателей оптимизации на основе роя частиц называет ее эволюционным алгоритмом [Shi и Eberhart, 1999]. Поскольку алгоритмы роевого интеллекта выполняются в том же общем виде, что и эволюционные алгоритмы, то есть путем эволюционного формирования популяции кандидатных решений задачи, которые улучшаются с каждой итерацией, мы рассматриваем роевой интеллект как эволюционный алгоритм.

Вся эта терминология неточна и контекстно зависима, однако в этой книге мы остановимся на термине *эволюционный алгоритм* и будем им обозначать алгоритм, который эволюционно формирует решение задачи в течение многочисленных итераций. Как правило, в соответствии со своей биологической основой одна итерация эволюционного алгоритма называется поколением. Однако это простое определение эволюционного алгоритма не является идеальным, потому что, например, оно подразумевает, что градиентный спуск – это эволюционный алгоритм, и никто не готов это признать. Таким образом, терминология эволюционных алгоритмов неоднородна и может вводить в заблуждение. Мы используем щекотливое определение, что алгоритм является эволюционным, если он в целом рассматривается как эволюционный алгоритм. Эта цикличность сначала надоедает, но те из нас, кто работает в данной области, через некоторое время к ней привыкают. В конце концов, естественный отбор определяется как выживание наиболее приспособленных, а под приспособленностью подразумеваются особи, которые с наибольшей вероятностью выживут.

## 1.2. Зачем нужна еще одна книга по эволюционным алгоритмам?

Существует ряд прекрасных книг по эволюционным алгоритмам, в которых поднимается вопрос, зачем нужен еще один учебник по теме эволюционных алгоритмов. Причина, почему эта книга была написана, состоит в том, чтобы предложить педагогический подход, перспективное видение и материал, которые недоступны ни в одной другой книге. В частности, мы надеемся, что эта книга предложит следующее.

- В книге дается простой восходящий подход, который помогает читателю получить четкое, но теоретически строгое понимание эволюционных алгоритмов. Во многих книгах обсуждаются различные эволюционные алгоритмы как алгоритмы из справочника без какой-либо теоретической поддержки. Другие книги больше похожи на научные монографии, чем на учебники, и не совсем доступны среднестатистическому студенту-инженеру. Эта книга пытается найти баланс, представляя простые в реализации алгоритмы наряду с небольшим объемом строгих теоретических выкладок и обсуждения компромиссов.
- В книге приведены простые примеры, которые дают читателю интуитивное понимание математики, уравнений и теории эво-

люционных алгоритмов. Во многих книгах представлена теория эволюционных алгоритмов, а затем приводятся примеры или задачи, которые не поддаются интуитивному пониманию. Вместе с тем вполне можно привести простые примеры и задачи, для решения которых нужны лишь бумага и карандаш. Эти простые задачи позволяют студенту непосредственнее увидеть то, как теория работает на практике.

- Исходный код на основе языка программирования MATLAB® для всех примеров в книге доступен на веб-сайте автора<sup>2</sup>. В ряде других учебников тоже предоставлен исходный код, но он часто является неполным или устаревшим, что расстраивает читателя. Адрес электронной почты автора также доступен на веб-сайте, и я с энтузиазмом приветствую отзывы, комментарии, предложения по улучшению исходного кода и его исправлениям. Конечно же, веб-адреса могут устареть, но эта книга содержит алгоритмические высокоуровневые листинги псевдокода, которые более постоянны, чем любые конкретные листинги программ. Обратите внимание, что примеры и программный код на MATLAB не предназначены в качестве эффективных или конкурентоспособных алгоритмов оптимизации; напротив, они предназначены только для того, чтобы дать читателю возможность получить общее представление о лежащих в их основе концепциях. Любое серьезное исследование или применение должно опираться на пример программного кода, который выступает только в качестве предварительной отправной точки.
- Данная книга включает в себя теорию и недавно разработанные эволюционные алгоритмы, которые недоступны в большинстве других учебников. Эти темы включают марковские теоретические модели эволюционных алгоритмов, системно-динамические модели эволюционных алгоритмов, алгоритмы на основе искусственной пчелиной семьи, биогеографическую оптимизацию, оппозиционное самообучение, алгоритмы на основе искусственных косячков рыб, перемешанных лягушачьих прыжков, оптимизацию на основе бактериальной кормодобычи и многие другие. Эти темы являются последними дополнениями в современное состояние дел в данной области, и их освещение в данной книге не соответствует уровню их освещения ни в одной другой книге.

<sup>2</sup> См. <http://academic.csuohio.edu/simond/EvolutionaryOptimization> – если адрес поменяется, то его легко можно отыскать в интернете.

Тем не менее эта книга не предназначена для того, чтобы быть высокоуровневым описанием текущего состояния дел в какой-либо конкретной области эволюционно-алгоритмических исследований. Вместо этого настоящая книга предназначена для высокоуровневого описания многих областей эволюционно-алгоритмических исследований, чтобы читатель мог получить широкое представление об эволюционных алгоритмах и иметь хорошие возможности для проведения дополнительных исследований в данной области в его текущем развитии.

### 1.3. Предварительные условия

В общем случае студент ничего не получит из такого курса без написания собственного программного продукта на основе эволюционного алгоритма. Поэтому в качестве предварительного условия можно было бы указать компетентные навыки программирования. В университете, где я преподаю этот курс студентам электротехники и вычислительной техники, конкретные предварительные условия для прохождения данного курса отсутствуют; предварительным условием для студентов старших курсов является статус старшекурсника, для студентов-выпускников предварительные условия отсутствуют. Однако я исхожу из того, что и старшекурсники, и студенты-выпускники являются хорошими программистами.

Используемые в книге обозначения предполагают, что читатель знаком со стандартной математической записью, которая используется в алгебре, геометрии, теории множеств и математическом анализе. Поэтому еще одним предварительным условием для понимания этой книги является уровень математической зрелости, характерный для продвинутого студента старших курсов. Математические обозначения описаны в разделе 1.5. Если читатель способен понять обозначения, описанные в указанном разделе, то существует неплохой шанс, что он сможет проследить за ходом обсуждения в оставшейся части книги.

Математика в теоретических разделах этой книги (глава 4, раздел 7.6, подавляющая часть главы 13 и несколько других разрозненных разделов) требует понимания теории вероятностей и линейных систем. Студенту будет трудно следить за логикой изложения этого материала, если он не закончил курс по этим двум предметам. В учебном курсе, ориентированном на старшекурсников, вероятно, следует этот материал пропустить.

## 1.4. Домашние задания

Задачи в конце каждой главы были написаны для того, чтобы обеспечить гибкость преподавателю и студенту. Задачи включают письменные и компьютерные упражнения. Письменные упражнения предназначены для укрепления студентом своего понимания теории, углубления своего интуитивного понимания понятий и развития своих аналитических навыков. Компьютерные упражнения призваны помочь студенту развить исследовательские навыки и научиться применять теорию к типам задач, которые обычно встречаются в промышленности. Оба типа задач важны для наращивания своей квалификации в работе с эволюционными алгоритмами. Разница между письменными и компьютерными упражнениями не является строгой, а скорее представляет собой нечеткое разделение. То есть некоторые письменные упражнения, возможно, потребуют некоторой работы с компьютером, а компьютерные упражнения потребуют некоторого анализа на бумаге. Инструктор может предусмотреть задания, связанные с эволюционными алгоритмами, исходя из собственных интересов. Семестровые, проектно-ориентированные задания часто являются наглядным примером таких тем, как эта. Например, студентам может быть поручено решить некоторую практическую оптимизационную задачу с использованием рассмотренных в данной книге эволюционных алгоритмов, применяя по одному эволюционному алгоритму на главу, а затем сравнивая результативность эволюционных алгоритмов и их вариации в конце семестра.

Справочник для преподавателей с решениями всех задач книги (как письменных, так и компьютерных упражнений) имеется в распоряжении у издателя. Преподавателям курсов рекомендуется обратиться к издателю за дополнительной информацией о получении справочника с решениями упомянутых задач. Для защиты целостности домашних заданий справочник с решениями задач будет предоставляться только преподавателям курса.

## 1.5. Обозначения

К сожалению, в английском языке нет гендерно-нейтрального местоимения третьего лица единственного числа. Поэтому мы используем слова «он» или «ему» для обозначения общего третьего лица, будь то мужчина или женщина. Такое именование может показаться неудобным как для писателей, так и для читателей, но оно представляется наиболее удовлетворительным выходом из трудного положения.

В приведенном ниже списке описаны некоторые математические обозначения, принятые в этой книге.

- $x \leftarrow y$  – вычислительное обозначение, указывающее, что переменной  $x$  присваивается значение  $y$ . Например, рассмотрим следующий ниже алгоритм:

$$a = \text{coefficient of } x^2$$

$$b = \text{coefficient of } x^1$$

$$c = \text{coefficient of } x^0$$

$$x^* \leftarrow (-b + \sqrt{b^2 - 4ac}) / (2a)$$

Первые три строки не являются инструкциями присваивания в алгоритме; они просто описывают или определяют значения  $a$ ,  $b$  и  $c$ . Эти три параметра могут быть заданы пользователем, другим алгоритмом или процессом. Последняя строка, однако, является инструкцией присваивания, которая обозначает, что значение в правой стороне от стрелки записывается в  $x^*$ .

- $df(\cdot)/dx$  – это полная производная от  $f(\cdot)$  по  $x$ . Например, предположим, что  $y = 2x$  и  $f(x, y) = 2x + 3y$ . Тогда  $f(x, y) = 8x$  и  $df(\cdot)/dx = 8$ .
- $f_x(\cdot)$ , также записываемое как  $\partial f(\cdot)/\partial x$ , является частной производной от  $f(\cdot)$  по  $x$ . Например, предположим, что  $y = 2x$  и  $f(x, y) = 2x + 3y$ . Тогда  $f_x(x, y) = 2$ .
- $\{x: x \in S\}$  – это множество всех  $x$  – таких, что  $x$  принадлежит множеству  $S$ . Аналогичная запись используется для обозначения тех значений  $x$ , которые удовлетворяют любому другому конкретному условию. Например, запись  $\{x: x^2 = 4\}$  тождественна  $\{x: x \in \{-2, +2\}\}$ , которая тождественна  $\{-2, +2\}$ .
- $[a, b]$  – это закрытый интервал между  $a$  и  $b$ , который означает, что  $\{x: a \leq x \leq b\}$ . В зависимости от контекста он может представлять множество целых чисел и множество вещественных чисел.
- $(a, b)$  – это открытый интервал между  $a$  и  $b$ , который означает, что  $\{x: a < x < b\}$ . В зависимости от контекста он может представлять множество целых чисел и множество вещественных чисел.
- Если из контекста понятно, что  $i \in S$ , то  $\{x_i\}$  является сокращенной записью для  $\{x_i: i \in S\}$ . Например, если  $i \in [1, N]$ , то  $\{x_i\} = \{x_1, x_2, \dots, x_N\}$ .



- $S_1 \cup S_2$  – это множество из всех  $x$  – таких, что  $x$  принадлежит либо множеству  $S_1$ , либо множеству  $S_2$ . Например, если  $S_1 = \{1, 2, 3\}$  и  $S_2 = \{7, 8\}$ , то  $S_1 \cup S_2 = \{1, 2, 3, 7, 8\}$ .
- $|S|$  – это число элементов во множестве  $S$ . Например, если  $S = \{i : i \in [4, 8]\}$ , то  $|S| = 5$ . Если  $S = \{3, 19, \pi, \sqrt{2}\}$ , то  $|S| = 4$ . Если  $S = \{\alpha : 1 < \alpha < 3\}$ , то  $|S| = \infty$ .
- $\emptyset$  – это пустое множество.  $|\emptyset| = 0$ .
- $x \bmod y$  – это остаток после деления  $x$  на  $y$ . Например,  $8 \bmod 3 = 2$ .
- $\lceil x \rceil$  – это верхнее округление  $x$ ; то есть наименьшее целое число, которое больше или равно  $x$ . Например,  $\lceil 3.9 \rceil = 4$ , а  $\lceil 5 \rceil = 5$ .
- $\lfloor x \rfloor$  – это нижнее округление  $x$ ; то есть наибольшее целое число, которое меньше или равно  $x$ , например  $\lfloor 3.9 \rfloor = 3$ , и  $\lfloor 5 \rfloor = 5$ .
- Запись  $\min_x f(x)$  обозначает задачу нахождения значения  $x$ , которое дает наименьшее значение  $f(x)$ . Кроме того, указанная запись может обозначать наименьшее значение  $f(x)$ . Например, предположим, что  $f(x) = (x - 1)^2$ . Тогда мы можем решить задачу  $\min_x f(x)$ , используя исчисление или построив график функции  $f(x)$  и визуально отметив наименьшее значение  $f(x)$ . Для этого примера мы находим, что  $\min_x f(x) = 0$ . Аналогичное определение справедливо для  $\max_x f(x)$ .
- $\arg \min_x f(x)$  – это значение  $x$ , которое приводит к наименьшему значению  $f(x)$ . Например, предположим, что  $f(x) = (x - 1)^2$ . Наименьшее значение  $f(x)$  равно 0, что происходит, когда  $x = 1$ , поэтому для данного примера  $\arg \min_x f(x) = 1$ . Аналогичное определение справедливо для  $\arg \max_x f(x)$ .
- $R^s$  – это множество всех вещественных  $s$ -элементных векторов. В зависимости от контекста эта запись может обозначать либо векторы-столбцы, либо векторы-строки.
- $R^{s \times p}$  – это множество всех вещественных  $s \times p$ -матриц.
- $\{y_k\}_{k=L}^U$  – это множество всех  $y_k$ , где целое число  $k$  колеблется между  $L$  и  $U$ . Например,  $\{y_k\}_{k=2}^5 = \{y_2, y_3, y_4, y_5\}$ .
- $\{y_k\}$  – это множество всех  $y_k$ , где целое число  $k$  колеблется между контекстно-зависимым нижним пределом и контекстно-зависимым верхним пределом. Например, предположим, что контекст указывает на наличие трех значений:  $y_1, y_2$  и  $y_3$ . Тогда  $\{y_k\} = \{y_1, y_2, y_3\}$ .

- Знак  $\exists$  означает «существует»; знак  $\nexists$  означает «не существует». Например, если  $Y = \{6, 1, 9\}$ , то  $\exists y < 2 : y \in Y$ . Однако  $\nexists y > 10 : y \in Y$ .
- Запись  $A \Rightarrow B$  означает, что  $A$  подразумевает  $B$ . Например,  $(x > 10) \Rightarrow (x > 5)$ .
- $I$  – это единичная матрица. Ее размеры зависят от контекста.

Дополнительные обозначения см. в списке сокращений на стр. 21.

## 1.6. План изложения

Настоящая книга разделена на шесть частей.

1. *Часть I* состоит из этого введения и еще одной главы, которая охватывает вводный материал, связанный с оптимизацией. Она знакомит с различными типами оптимизационных задач, простым, но эффективным алгоритмом восхождения к вершине холма и завершается обсуждением признаков, которые делают алгоритм интеллектуальным.
2. *Часть II* посвящена четырем эволюционным алгоритмам, которые обычно считаются классикой:
  - генетический алгоритм;
  - эволюционное программирование;
  - эволюционные стратегии;
  - генетическое программирование.

Часть II также включает главу, в которой рассматриваются методы математического анализа генетических алгоритмов. Часть II завершается главой, в которой обсуждаются некоторые из многих алгоритмических вариаций, которые могут использоваться в указанных классических алгоритмах. Эти же вариации можно использовать и в более поздних эволюционных алгоритмах, которые рассматриваются в следующей части.

3. В *части III* обсуждается несколько более поздних эволюционных алгоритмов. Некоторые из них не совсем новые и относятся к 1980-м годам, однако другие датируются только первым десятилетием XXI века.
4. В *части IV* обсуждаются специальные типы оптимизационных задач и показан способ модификации эволюционных алгоритмов

предыдущих глав для их решения. Эти специальные типы задач включают в себя:

- комбинаторные задачи, область решения которых состоит из целых чисел;
- ограниченные задачи, область решения которых ограничена известным множеством;
- многокритериальные задачи, в которых желательно минимизировать более одной целевой функции одновременно;
- задачи с шумными или дорогостоящими функциями приспособленности, для которых трудно получить точную результативность кандидатного решения или для которых вычислительно затратно оценить результативность кандидатного решения.

5. *Часть V* включает несколько приложений, в которых обсуждаются важные или интересные темы.

- *Приложение A* содержит различные практические советы для студентов и исследователей эволюционных алгоритмов.
- В *приложении B* обсуждается теорема об отсутствии бесплатных обедов, которая постулирует, что в среднем все оптимизационные алгоритмы работают одинаково. В нем также обсуждается вопрос о том, как использовать статистику для оценивания различий между эволюционными алгоритмами.
- В *приложении C* приведено несколько стандартных эталонных функций, которые можно использовать для сравнения результативности разных эволюционных алгоритмов.

## 1.7. Учебный курс на основе данной книги

Любой курс на основе настоящей книги должен начинаться с глав 1 и 2, в которых дается краткий обзор оптимизационных задач. Остальные главы после них можно изучать практически в любом порядке, в зависимости от предпочтений и интересов инструктора. Очевидным исключением является то, что изучение генетических алгоритмов (глава 3) должно предшествовать изучению их математических моделей (глава 4).

Кроме того, по крайней мере, одна глава в частях II или III (то есть, по крайней мере, один конкретный эволюционный алгоритм) должна быть подробно рассмотрена перед любой из глав в части IV.

Большинство курсов, как минимум, охватывает главы 3 и 5–7, для того чтобы предоставить студенту общие сведения в области классических

эволюционных алгоритмов. Если студенты обладают достаточно высоким уровнем математических знаний и если есть время, то курс также в какой-то момент должен включить главу 4. Глава 4 важна для студентов-выпускников, потому что она помогает им понять, что эволюционные алгоритмы – это не только качественный предмет, но для них также может и должна иметься некая теоретическая основа. Слишком много исследовательских работ по эволюционным алгоритмам сегодня основано на незначительных алгоритмических корректировках без какой-либо математической поддержки. Многие практикующие специалисты в области эволюционных алгоритмов интересуются только получением результатов, что хорошо, однако академические исследователи должны быть вовлечены как в теорию, так и в практику.

Главы в частях III и IV могут быть рассмотрены с учетом конкретных интересов преподавателя или студентов.

Приложения не включены в основную часть книги, потому что они не об эволюционных алгоритмах как таковых, но важность приложений не следует недооценивать. В частности, материалы в приложениях B и C имеют решающее значение и должны быть включены в каждый курс по эволюционным алгоритмам. Я рекомендую обсудить некоторые детали этих двух приложений сразу же после первой главы в частях II или III.

Собрав все приведенные выше советы вместе, предлагаем схематичный план для одного семестра старшего курса.

- Главы 1 и 2.
- Глава 3.
- Приложения B и C.
- Главы 4–8. Я рекомендую пропустить главу 4 для большинства студентов старших курсов и в случае коротких курсов.
- Несколько глав в части III, исходя из предпочтений инструктора. Рискую начать «эволюционно-алгоритмическую войну» с моими читателями, выскажу малопопулярную точку зрения и заявлю, что алгоритмы оптимизации на основе муравьиной кучи (ACO), оптимизации на основе роя частиц (PSO) и дифференциальной эволюции (DE) являются одними из наиважнейших «других» эволюционных алгоритмов, и поэтому инструктор должен как минимум охватить главы 10–12.
- Несколько глав в части IV, исходя из предпочтений инструктора и имеющегося времени.

# Глава 2

## Оптимизация

*Оптимизацией пронизано все то, что мы делаем, и она управляет почти каждым аспектом инженерного дела.*

– Деннис Бернштейн [Bernstein, 2006]

Как указано в приведенной выше цитате, оптимизация является частью почти всего, что мы делаем. Оптимизации подлежат штатные расписания, стили преподавания, экономические системы, игровые стратегии, биологические системы и системы здравоохранения. Оптимизация является увлекательной областью исследования не только из-за ее алгоритмического и теоретического содержания, но и из-за ее универсальной применимости.

### Краткий обзор главы

В данной главе дается краткий обзор оптимизации (раздел 2.1), включая ограниченную оптимизацию с учетом ограничений (раздел 2.2), оптимизационные задачи, которые имеют несколько целей (раздел 2.3) и оптимизационные задачи, которые имеют несколько решений (раздел 2.4). Подавляющая часть нашей работы в этой книге сосредоточена на непрерывных оптимизационных задачах, то есть задачах, где независимая переменная может варьироваться непрерывно. Однако задачи, в которых независимая переменная ограничена конечным множеством, то есть так называемые комбинаторные задачи, также представляют большой интерес, и мы познакомимся с ними в разделе 2.5. В разделе 2.6 мы представим простой универсальный оптимизационный алгоритм под названием «восхождение к вершине холма», а также обсудим несколько его вариаций. Наконец, в разделе 2.7 мы обсудим несколько идей, связанных с природой интеллекта, и покажем, как они связаны с

алгоритмами эволюционной оптимизации, которые мы представим в последующих главах.

## 2.1. Неограниченная оптимизация

Оптимизация применима практически ко всем сферам жизни. Оптимизационные алгоритмы могут применяться ко всему, от разведения трубокзубов (африканских муравьедов) до исследования зигот (оплодотворенной яйцеклетки). Возможные применения эволюционных алгоритмов ограничены только воображением инженера, именно по этой причине в течение прошедших нескольких десятилетий эволюционные алгоритмы очень широко исследовались и применялись на практике.

Например, в инженерном деле эволюционные алгоритмы используются для поиска лучших траекторий робота для определенной задачи. Предположим, что на вашем производственном предприятии есть робот, и вы хотите, чтобы он выполнял свою задачу таким образом, чтобы он заканчивал ее как можно быстрее или использовал наименьшую возможную мощность. Как рассчитать лучший возможный путь для робота? Существует так много возможных путей, что задача найти лучшее решение является чрезвычайно сложной. Однако эволюционные алгоритмы способны сделать такую задачу управляемой (если вообще не легкой) и, по крайней мере, найти хорошее решение (если вообще не лучшее). Роботы очень нелинейны, поэтому, как мы видели в простом примере, представленном ранее в этой главе, поисковое пространство для задач робототехнической оптимизации сводится к большому числу пиков и впадин. Но в случае с робототехническими задачами ситуация еще хуже, потому что пики и впадины лежат в многомерном пространстве (а не в простом трехмерном пространстве, которое мы видели ранее). Сложность задач робототехнической оптимизации делает их естественной мишенью для эволюционных алгоритмов. Эта идея была применена как для стационарных роботов (то есть роботов-манипуляторов), так и для мобильных роботов.

Эволюционные алгоритмы также используются для тренировки нейронных сетей и систем нечеткой логики. В нейронных сетях мы должны выяснить архитектуру сети и нейронные веса, с тем чтобы получить лучшую результативность сети. Опять же, существует так много возможностей, что эта задача просто колоссальна. Вместе с тем можно применить эволюционные алгоритмы и найти лучшую конфигурацию и лучшие веса. Та же задача связана и с системами нечеткой логики. Какую базу правил мы должны использовать? Сколько функций принад-

лежности мы должны использовать? Какие формы функций принадлежности следует использовать? Эволюционные алгоритмы помогали и помогают решать эти сложные оптимизационные задачи.

Эволюционные алгоритмы также применялись в медицинской диагностике. Например, после биопсии специалисты-медики идентифицируют, какие клетки являются раковыми, а какие нет. Какие признаки следует искать, чтобы диагностировать рак? Какие признаки являются наиболее важными, а какие – неактуальными? Какие признаки важны только, если пациент относится к определенной демографической группе? Эволюционный алгоритм способен помогать принимать такие решения. Эволюционному алгоритму всегда нужен профессионал, который его запустит и натренирует, но после этого он фактически может превзойти своего учителя. Эволюционный алгоритм не только не устает и не истощается, но и может извлекать шаблоны из данных, которые могут быть слишком утонченными, чтобы их могли распознавать люди. Эволюционные алгоритмы применялись для диагностики нескольких разных видов рака.

После того как болезнь была диагностирована, следующий трудный вопрос включает в себя управление болезнью. Например, после обнаружения рака какое лечение для пациента является лучшим? Как часто следует проводить облучение, каким оно должно быть и в каких дозах? Как следует лечить побочные эффекты? Это еще одна сложная оптимизационная задача. Неправильное лечение может принести больше вреда, чем пользы. Определение правильного вида лечения является нетривиальной функцией таких факторов, как тип рака, локализация рака, демографические показатели, общее состояние здоровья и другие. Поэтому генетические алгоритмы применяются не только для диагностики заболеваний, но и для планирования лечения.

Эволюционные алгоритмы следует рассматривать всегда, когда вы хотите решить сложную задачу. Это не означает, что эволюционные алгоритмы всегда являются лучшим выбором для работы. В калькуляторах программа на основе эволюционного алгоритма для сложения чисел не используется, потому что для этого есть гораздо более простые и эффективные алгоритмы. Однако эволюционные алгоритмы, по крайней мере, следует рассматривать для любой нетривиальной задачи. Если вы хотите разработать проект объекта жилищного строительства или транспортной системы, то эволюционный алгоритм, возможно, будет ответом. Если вы хотите спроектировать сложную электрическую схему или компьютерную программу, то эволюционный алгоритм вполне сможет выполнить эту работу.

Оптимизационная задача может быть записана как минимизационная или как максимизационная задача. Иногда мы пытаемся минимизировать функцию, а иногда пытаемся ее максимизировать. Эти две задачи легко преобразуются в другую форму:

$$\begin{aligned}\min_x f(x) &\Leftrightarrow \max_x [-f(x)] \\ \max_x f(x) &\Leftrightarrow \min_x [-f(x)].\end{aligned}\tag{2.1}$$

Функция  $f(x)$  называется целевой функцией, а вектор  $x$  – независимой переменной, или переменной решения. Обратите внимание, что в зависимости от контекста термины независимая переменная и переменная решения иногда относятся ко всему вектору  $x$ , а иногда к определенным элементам в  $x$ . Элементы вектора  $x$  также называются признаками решения. Число элементов в  $x$  называется размерностью задачи. Как мы видим из уравнения (2.1), любой алгоритм, разработанный для минимизации функции, может быть легко использован для максимизации функции, и любой алгоритм, разработанный для максимизации функции, может быть легко использован для ее минимизации. Когда мы пытаемся минимизировать функцию, мы называем значение этой функции функцией стоимости, или просто стоимостью (от англ. *cost*). Когда мы пытаемся максимизировать функцию, то мы называем значение этой функции приспособленностью (от англ. *fitness*).

$$\begin{aligned}\min_x f(x) \Rightarrow f(x) &\text{ называется «стоимостью»,} \\ &\text{или «целевым критерием»} \\ \max_x f(x) \Rightarrow f(x) &\text{ называется «приспособленностью»,} \\ &\text{или «целевым критерием»}.\end{aligned}\tag{2.2}$$

### ■ Пример 2.1

Данный пример иллюстрирует терминологию, которую мы используем в этой книге. Предположим, мы хотим минимизировать функцию

$$f(x, y, z) = (x - 1)^2 + (y + 2)^2 + (z - 5)^2 + 3.\tag{2.3}$$

Переменные  $x$ ,  $y$  и  $z$  называются независимыми переменными, переменными решения или признаками решения; все три термина эквивалентны. Это трехмерная задача,  $f(x, y, z)$  называется целевой функцией, или функцией стоимости. Мы можем изменить задачу на задачу максимизации, определив  $g(x, y, z) = -f(x, y, z)$ , и попытаться максимизировать  $g(x, y, z)$ . Функция  $g(x, y, z)$  называется целевой функцией, или функцией приспособленности. Решение задачи  $\min f(x, y, z)$  такое же, как и реше-



ние задачи  $\max u(x, y, z)$ , и равно  $x = 1$ ,  $y = -2$  и  $z = 5$ . Однако оптимальное значение для  $f(x, y, z)$  является противоположным оптимальному значению для  $g(x, y, z)$ .

Иногда оптимизация проста и может быть выполнена с помощью аналитических методов, как мы увидим в следующем далее примере.

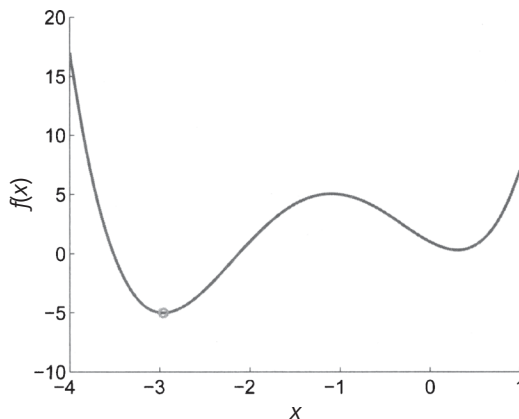
### ■ Пример 2.2

Рассмотрим задачу

$$\min_x f(x), \quad \text{где } f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1. \quad (2.4)$$

График  $f(x)$  показан на рис. 2.1. Поскольку функция  $f(x)$  является квадратичным полиномом (так называемым многочленом четвертой степени, или четвертого порядка), мы знаем, что она имеет не более трех стационарных точек, то есть три значения  $x$ , при которых ее производная  $f'(x) = 0$ . Эти точки, как видно по рис. 2.1, встречаются при  $x = -2.96$ ,  $x = -1.10$  и  $x = 0.31$ . Мы можем подтвердить, что  $f'(x)$ , которая равна  $4x^3 + 15x^2 + 8x - 4$ , равна нулю при этих трех значениях  $x$ . Далее мы можем найти, что вторая производная от  $f(x)$  в этих трех точках равняется

$$f''(x) = 12x^2 + 30x + 8 = \begin{cases} 24.33, & x = -2.96 \\ -10.48, & x = -1.10 \\ 18.45, & x = 0.31 \end{cases} \quad (2.5)$$



**Рис. 2.1.** Пример 2.2: простая задача минимизации.  $f(x)$  имеет два локальных минимума и один глобальный минимум, который встречается при  $x = -2.96$

Напомним, что вторая производная функции при локальном минимуме положительна, а вторая производная функции при локальном максимуме отрицательна. Таким образом, значения  $f''(x)$  в стационарных точках подтверждают, что  $x = -2.96$  является локальным минимумом,  $x = -1.10$  – локальным максимумом, а  $x = 0.31$  – еще одним локальным минимумом. ■

Функция примера 2.2 имеет два локальных минимума и один глобальный минимум. Обратите внимание, что глобальный минимум также является локальным минимумом. В некоторых функциях  $\min_x f(x)$  встречается при более чем одном значении  $x$ ; если это происходит, то  $f(x)$  имеет несколько глобальных минимумов. Локальный минимум  $x^*$  может быть определен как

$$f(x^*) < f(x) \text{ для всех } x \text{ such that } \|x - x^*\| < \varepsilon, \quad (2.6)$$

где  $\|\cdot\|$  – это некий метрический показатель расстояния, а  $\varepsilon > 0$  – некий задаваемый пользователем размер окрестности. На рис. 2.1 видно, что  $x = 0.31$  является локальным оптимумом, если, например, размер окрестности  $\varepsilon = 1$ , но не локальным оптимумом, если  $\varepsilon = 4$ . Глобальный минимум  $x^*$  может быть определен как

$$f(x^*) \leq f(x) \text{ для всех } x. \quad (2.7)$$

## 2.2. Ограниченная оптимизация

Довольно часто оптимизационная задача имеет ограничения. То есть перед нами поставлена задача минимизации некой функции  $f(x)$  с ограничениями на допустимые значения  $x$ , как в следующем ниже примере.

### ■ Пример 2.3

Рассмотрим задачу

$$\min_x f(x), \quad \text{где } f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1 \quad (2.8)$$

и  $x \geq -1.5$ .

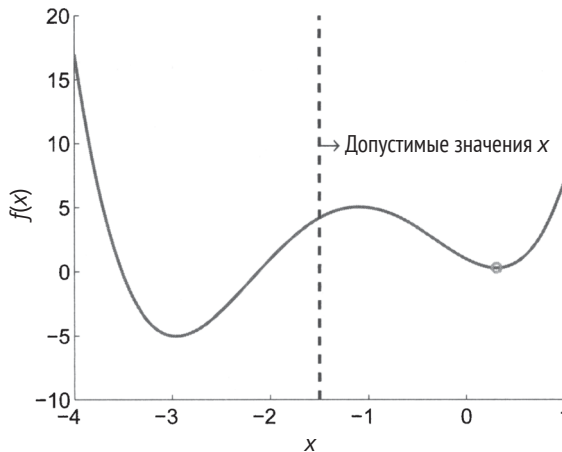
Это та же задача, что и в примере 2.2, за исключением того, что  $x$  ограничено. График  $f(x)$  и допустимые значения  $x$  показаны на рис. 2.2, и анализ графика показывает ограниченный минимум. Чтобы решить эту задачу аналитически, мы находим три стационарные точки  $f(x)$ , как в примере 2.2, игнорируя ограничение. Мы находим, что два локаль-

ных минимума встречаются при  $x = -2.96$  и  $x = 0.31$ , как в примере 2.2. Мы видим, что только одно из этих значений,  $x = 0.31$ , удовлетворяет ограничению. Далее мы должны вычислить  $f(x)$  на границе ограничения, чтобы убедиться, что она меньше, чем при локальном минимуме  $x = 0.31$ . Мы находим, что

$$f(x) = \begin{cases} 4.19, & \text{для } x = -1.50 \\ 0.30, & \text{для } x = 0.30 \end{cases} \quad (2.9)$$

Мы видим, что  $x = 0.31$  является минимизирующим значением  $x$  для задачи ограниченной минимизации.

Если бы граница ограничения была дальше влево, то минимизирующее значение  $x$  возникло бы на границе ограничения, а не на локальном минимуме  $x = 0.31$ . Если бы граница ограничения была слева от  $x = -2.96$ , то минимизирующее значение  $x$  для задачи ограниченной минимизации было бы таким же, как и для задачи неограниченной минимизации.



**Рис. 2.2.** Пример 2.3: простая задача ограниченной минимизации. Ограниченный минимум имеет место при  $x = 0.31$

■

Реальные оптимизационные задачи почти всегда имеют ограничения. Кроме того, в реальных оптимизационных задачах оптимизирующее значение независимой переменной почти всегда возникает на границе ограничения. Это неудивительно, потому что мы обычно рассчитываем получить лучший инженерный проект, распределение ре-

сурсов или другую цель оптимизации, используя всю доступную энергию или силу или какой-либо другой ресурс [Bernstein, 2006]. Поэтому ограничения важны практически во всех реальных оптимизационных задачах. В главе 19 более подробно рассматривается ограниченная эволюционная оптимизация.

## 2.3. Многокритериальная оптимизация

Реальные оптимизационные задачи не только ограничены, но и являются многокритериальными<sup>1</sup>. Это означает, что мы заинтересованы в минимизации более чем одной меры одновременно. Например, в задаче управления двигателем мы, возможно, будем заинтересованы в минимизации ошибки слежения и одновременно минимизации энергопотребления. Мы можем получить очень небольшую ошибку слежения за счет высокого энергопотребления либо можем допустить большую ошибку слежения при малом энергопотреблении. В крайнем случае, мы можем выключить двигатель, для того чтобы достигнуть нулевого энергопотребления, но тогда наша ошибка слежения не будет очень хорошей.

### ■ Пример 2.4

Рассмотрим задачу

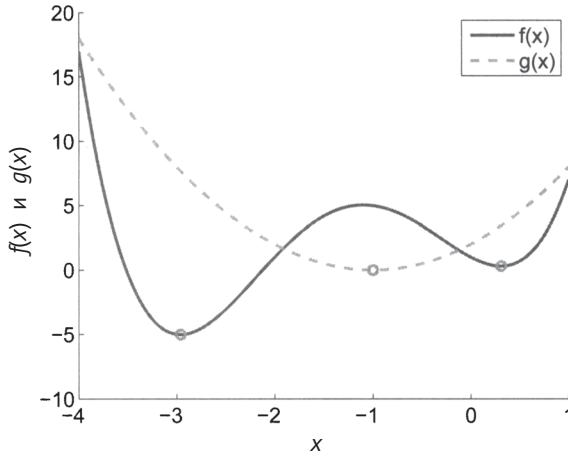
$$\min_x [f(x) \text{ и } g(x)], \quad \text{где } f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1 \quad (2.10)$$

$$\text{и } g(x) = 2(x + 1)^2.$$

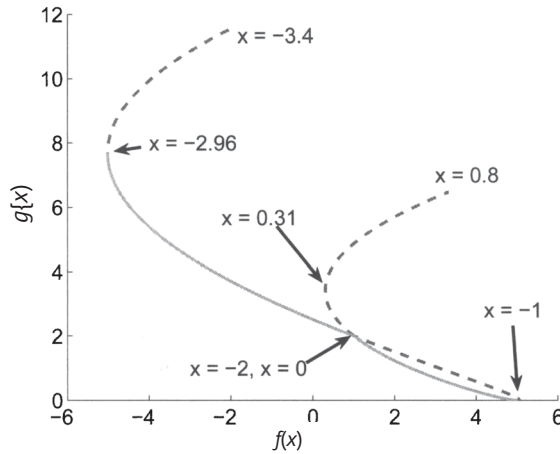
Первый целевой критерий минимизации,  $f(x)$ , такой же, как и в примере 2.2. Однако теперь мы также хотим минимизировать  $g(x)$ . График  $f(x)$ ,  $g(x)$  и их минимумы показаны на рис. 2.3. Анализ графика показывает, что  $x = -2.96$  минимизирует  $f(x)$ , в то время как  $x = -1$  минимизирует  $g(x)$ . Не ясно, какое значение  $x$  будет для этой задачи наиболее предпочтительным, потому что у нас две противоречивые цели. Однако по рис. 2.3 должно быть очевидно, что мы никогда не захотим, чтобы  $x < -2.96$  или  $x > 0.31$ . Если  $x$  уменьшается с  $-2.96$  или увеличивается с  $0.31$ , то целевые критерии  $f(x)$  и  $g(x)$  увеличиваются, что явно нежелательно.

Один из способов вычислить эту задачу – построить график  $g(x)$  как функцию от  $f(x)$ . Это показано на рис. 2.4, где мы варьировали  $x$  от  $-3.4$  до  $0.8$ . Ниже мы рассмотрим каждый раздел графика.

<sup>1</sup> Такие оптимизационные задачи также называются многоцелевыми, многообъектными и многоатрибутными. – *Прим. перев.*



**Рис. 2.3.** Пример 2.4: простая задача многокритериальной минимизации.  $f(x)$  имеет два минимума, а  $g(x)$  – один минимум. Два целевых критерия конфликтуют



**Рис. 2.4.** Пример 2.4: на этом рисунке  $g(x)$  показана как функция от  $f(x)$  для простой задачи многокритериальной минимизации по мере того, как  $x$  варьируется от  $-3.4$  до  $0.8$ . Сплошная линия – это фронт Парето

- $x \in [-3.4, -2.96]$ : когда  $x$  увеличивается от  $-3.4$  до  $-2.96$ ,  $f(x)$  и  $g(x)$  уменьшаются вместе. Поэтому мы никогда не выберем  $x < -2.96$ .
- $x \in [-2.96, -1]$ : когда  $x$  увеличивается от  $-2.96$  до  $-1$ ,  $g(x)$  уменьшается, а  $f(x)$  увеличивается.

- $x \in [-1, 0]$ : когда  $x$  увеличивается от  $-1$  до  $0$ ,  $g(x)$  увеличивается, а  $f(x)$  уменьшается. Однако на этой части графика, несмотря на то что  $g(x)$  увеличивается, он все еще меньше  $g(x)$  для  $x \in [-2, -1]$ , поэтому  $x \in [-1, 0]$  предпочтительнее  $x \in [-2, -1]$ .
- $x \in [0, 0.31]$ : когда  $x$  увеличивается от  $0$  до  $0.31$ ,  $g(x)$  увеличивается, а  $f(x)$  уменьшается. По графику мы видим, что для  $x \in [0, 0.31]$   $g(x)$  больше, чем она на части графика  $x \in [-2.96, -2]$ . Поэтому мы никогда не захотим выбрать  $x \in [0, 0.31]$ .
- $x \in [0.31, 0.8]$ : наконец, когда  $x$  увеличивается от  $0.31$  до  $0.8$ ,  $f(x)$  и  $g(x)$  увеличиваются вместе. Поэтому мы никогда не выберем  $x > 0.31$ .

Резюмируя вышеизложенные результаты, мы начертим потенциально желательные значения  $f(x)$  и  $g(x)$  сплошной линией на рис. 2.4. Для значений  $x$  на сплошной линии мы не можем найти никаких других значений  $x$ , которые уменьшат одновременно  $f(x)$  и  $g(x)$ . Сплошная линия называется фронтом Парето, а соответствующее множество значений  $x$  – множеством Парето.

$$\begin{aligned} \text{множество Парето: } \quad x^* &= \{x : x \in [-2.96, -2] \text{ или } x \in [-1, 0]\}; \\ \text{фронт Парето: } \quad &\{(f(x), g(x)) : x \in x^*\}. \end{aligned} \quad (2.11)$$

После того как мы получим множество Парето, мы больше ничего не можем сказать об оптимальном значении  $x$ . Выбор точки вдоль фронта Парето теперь является вопросом инженерного суждения как окончательного решения. Фронт Парето дает множество разумных вариантов на выбор, но любой вариант  $x$  из множества Парето по-прежнему влечет за собой компромисс между двумя целевыми критериями.

■

Пример 2.4 представляет собой довольно простую задачу многокритериальной оптимизации только с двумя целевыми критериями. Типичная оптимизационная задача в реальном мире включает в себя гораздо больше, чем просто две цели, и поэтому ее фронт Парето трудно получить. Даже если бы мы могли получить фронт Парето, мы не смогли бы визуализировать его из-за его высокой размерности. В главе 20 более подробно рассматривается эволюционная многокритериальная оптимизация.