Содержание

Введение

В настоящее время применение прикладных моделей оптимизации становится все более распространенным в различных областях, таких как инженерия, экономика, биология, искусственный интеллект и другие. Эти модели представляют собой мощный инструмент для решения разнообразных задач, таких как поиск оптимальных параметров, минимизация функций стоимости или максимизация производительности систем.

1. Актуальность

Актуальность изучения прикладных моделей оптимизации обусловлена растущим интересом к разработке эффективных алгоритмов решения сложных задач в условиях современного информационного общества. Теоретическая и практическая значимость этих моделей подтверждается их широким применением в различных областях науки и техники.

2. Цель

Целью настоящей работы является исследование и сравнительный анализ эволюционных алгоритмов оптимизации, таких как Covariance Matrix Adaptation Evolution Strategy (CMA-ES) и Non-dominated Sorting Genetic Algorithm (NSGA), в их различных вариантах (NSGA-I, NSGA-II, NSGA-III).

3. Задачи

Для достижения указанной цели необходимо:

- Изучить теоретические основы эволюционных алгоритмов оптимизации, включая CMA-ES и NSGA.

- Разработать программную реализацию алгоритмов CMA-ES и NSGA-I-II-III на выбранном языке программирования.

- Провести сравнительный анализ производительности и эффективности этих алгоритмов на ряде стандартных тестовых задач оптимизации.

- Выявить основные отличия между NSGA-I, NSGA-II и NSGA-III в

контексте их применения к различным задачам оптимизации.

4. Объект и предмет исследования

Объектом исследования являются эволюционные алгоритмы оптимизации, а предметом исследования - анализ их эффективности и применимости.

5. Теоретическая основа

Теоретическая основа исследования включает работы в области эволюционных алгоритмов оптимизации, таких как работы о CMA-ES и NSGA, а также труды по сравнительному анализу этих алгоритмов.

6. Методы исследования

Методы исследования включают в себя анализ литературных источников, разработку программного кода, проведение экспериментов и статистический анализ результатов.

7. Новизна

Новизна настоящего исследования заключается в сравнительном анализе эволюционных алгоритмов оптимизации CMA-ES и NSGA в их различных вариантах, а также в выявлении основных отличий между ними.

8. Практическая значимость

Результаты исследования могут быть полезны для разработчиков алгоритмов оптимизации при выборе наиболее подходящего метода для конкретной задачи, а также для исследователей, занимающихся оптимизацией систем различного назначения.

9. Структура работы

Работа состоит из введения, теоретической части, описания методологии и экспериментов, анализа результатов и заключения.

Глава 1: Теоретические основы эволюционных алгоритмов оптимизации

1.1. Основные понятия и принципы эволюционных алгоритмов

Эволюционные алгоритмы (ЭА) представляют собой мощный класс алгоритмов оптимизации, инспирированных принципами естественного отбора и генетики. Они основаны на идее эволюции и принципах естественного отбора, где происходит многократное повторение процессов селекции, скрещивания и мутации для нахождения наиболее приспособленных решений к поставленной задаче. Популяция, состоящая из набора потенциальных решений (индивидов), подвергается постоянным изменениям и улучшениям через эти операторы.

Применение эволюционных алгоритмов началось в 1960-х годах, когда Джон Холланд и его коллеги впервые предложили Генетический Алгоритм (ГА). С тех пор эти методы стали одними из наиболее широко используемых в области оптимизации в различных областях, включая инженерию, финансы, биологию, медицину и многие другие. Их применение возможно благодаря их способности работать с большим количеством параметров и эффективно искать оптимальные решения в сложных пространствах поиска.

Процесс оптимизации включает в себя последовательное применение операторов селекции, скрещивания и мутации к популяции индивидов. Оператор селекции выбирает индивидов с высокой приспособленностью для перехода в следующее поколение, оператор скрещивания комбинирует характеристики двух или более индивидов для создания новых потенциальных решений, а оператор мутации вносит случайные изменения в индивидов для разнообразия и сохранения исследования пространства решений.

1.2. Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) был предложен Николасом Хансоном в 1996 году и с тех пор стал одним из наиболее

эффективных и широко применяемых методов оптимизации с непрерывными переменными. Алгоритм CMA-ES представляет собой высокоадаптивный эволюционный метод, способный эффективно решать широкий класс задач оптимизации с непрерывными параметрами. Он базируется на идее адаптивной ковариационной матрицы, которая управляет процессом изменения распределения точек в пространстве параметров, обеспечивая более эффективный поиск оптимальных решений.

Основные компоненты CMA-ES включают в себя эволюционную стратегию для адаптации параметров распределения, которая позволяет алгоритму настраивать свои внутренние параметры в соответствии с характеристиками задачи. Кроме того, CMA-ES использует метод адаптивного выбора размера шага для эффективного поиска по пространству параметров, что позволяет учитывать различные характеристики локальной структуры функции оптимизации. Механизмы адаптации шага мутации и ковариационной матрицы дополнительно повышают эффективность алгоритма, позволяя ему адаптироваться к разнообразным условиям задачи.

CMA-ES нашел широкое применение в различных областях, включая машинное обучение, где его эффективность используется для настройки параметров моделей и оптимизации функций потерь, оптимизацию процессов в промышленности, где его применяют для оптимизации технологических процессов и управления ресурсами, а также в алгоритмической торговле, где его используют для оптимизации торговых стратегий и портфелей.

1.3. Non-dominated Sorting Genetic Algorithm (NSGA)

NSGA (Non-dominated Sorting Genetic Algorithm) является эволюционным алгоритмом, который был разработан Кришна Кумаром Дебом в 2000 году специально для решения сложных задач многокритериальной оптимизации. Он представляет собой мощный инструмент для нахождения набора непревосходящих решений, обладающих оптимальными значениями по всем заданным критериям одновременно.

Принцип работы NSGA основан на инновационной концепции сортировки по недоминированию. Этот подход позволяет эффективно выделять решения, которые не могут быть улучшены ни по одному критерию без ухудшения по другим. Таким образом, NSGA обеспечивает равномерное покрытие фронта Парето - множества всех недоминируемых решений, представляющих оптимальные компромиссы между различными целями или критериями.

Помимо процедуры сортировки по недоминированию, NSGA использует разнообразные механизмы для поддержания многообразия в популяции. Это важно для обеспечения достаточного покрытия пространства поиска и предотвращения преждевременной сходимости к локальным оптимумам.

NSGA и его вариации, такие как NSGA-II и NSGA-III, нашли применение в широком спектре областей, включая проектирование механических систем, управление энергопотреблением, оптимизацию производственных процессов и дизайн многокритериальных систем в области инженерии, экономики, экологии и других. Их успешное применение подтверждается многочисленными исследованиями и практическими примерами, демонстрируя их эффективность и универсальность в решении сложных задач оптимизации.

1.4. Сравнительный анализ CMA-ES и NSGA

Сравнительный анализ алгоритмов CMA-ES и NSGA позволяет более глубоко понять их принципы работы, эффективность, сложность настройки и области применения.

1.4.1. Принципы работы

CMA-ES и NSGA основаны на различных принципах работы, что влияет на их эффективность в решении различных задач оптимизации. CMA-ES использует адаптивную ковариационную матрицу для эффективного поиска оптимальных решений в пространстве параметров. Это позволяет алгоритму быстро адаптироваться к сложным ландшафтам функций и находить глобальные оптимумы. С другой стороны, NSGA основан на сортировке по

недоминированию, что позволяет поддерживать множество недоминируемых решений в популяции. Это делает NSGA эффективным для решения задач многокритериальной оптимизации, где требуется нахождение компромиссных решений.

1.4.2. Эффективность поиска

Эффективность поиска оптимальных решений зависит от природы задачи и характеристик алгоритмов. CMA-ES обычно считается более эффективным для задач с непрерывными переменными и гладкими функциями, благодаря своей способности адаптироваться к сложным ландшафтам функций. С другой стороны, NSGA лучше подходит для задач многокритериальной оптимизации, где требуется нахождение набора недоминируемых решений. Он способен поддерживать разнообразие решений на фронте Парето и находить компромиссные решения между различными критериями.

1.4.3. Сложность и требования к настройке

Сравнительный анализ также включает оценку сложности и требований к настройке каждого алгоритма. CMA-ES, обычно, имеет меньшее количество параметров и проще в настройке, что делает его более привлекательным выбором для практических приложений. Однако, NSGA имеет более сложную структуру и требует более тщательной настройки параметров, особенно при решении задач с большим числом критериев или сложных функций приспособленности.

1.4.4. Применимость

Оценка областей применения каждого алгоритма является важным аспектом сравнительного анализа. Оба алгоритма находят применение в различных областях, но их выбор зависит от конкретной задачи и требований к оптимизации. Например, CMA-ES может быть более эффективным для задач оптимизации параметров в машинном обучении, где требуется быстрый и точный поиск оптимальных параметров модели. С другой стороны, NSGA может быть предпочтительным выбором для проектирования

многокритериальных систем, таких как инженерные конструкции или управление ресурсами, где необходимо учитывать несколько конфликтующих целей.

1.5. Выводы

Исходя из проведенного сравнительного анализа, можно сделать вывод о том, что как CMA-ES, так и NSGA имеют свои преимущества и недостатки, и выбор конкретного метода зависит от характера задачи оптимизации. Более глубокое понимание принципов работы, эффективности и требований к настройке каждого алгоритма позволяет выбирать наиболее подходящий метод для конкретной задачи оптимизации.

Глава 2: Расчётно-аналитический анализ алгоритмов CMA-ES и NSGA

2.1. Постановка задачи и выбор программного обеспечения

Для проведения расчётно-аналитического анализа были выбраны задачи оптимизации функций Розенброка и Зиделя, а также задачи многокритериальной оптимизации с использованием функций DTLZ. Для реализации алгоритмов и проведения экспериментов выбрано программное обеспечение Python с библиотеками DEAP (Distributed Evolutionary Algorithms in Python) и pymoo. Эти библиотеки предоставляют мощные инструменты для реализации эволюционных алгоритмов и анализа их производительности.

2.2. Реализация алгоритмов CMA-ES и NSGA на Python

В этом разделе представлена реализация алгоритмов CMA-ES и NSGA-I, NSGA-II и NSGA-III на языке Python. Полные исходные коды приведены в приложении 1. В данном разделе приводится словесное описание их реализации.

2.2.1. Реализация и экспериментирование с CMA-ES

Задача оптимизации функции Розенброка

Функция Розенброка, также известная как "банановая функция", является стандартной тестовой функцией в оптимизации. Она определяется следующим образом:

$$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$$

Рисунок 1 – функция Розенброка

Эта функция имеет глобальный минимум в точке

x = (1,1,…,1)

x=(1,1,…,1), где значение функции равно нулю. Функция Розенброка выбрана

для эксперимента по нескольким причинам:

Сложность ландшафта: Функция содержит узкие, изогнутые долины, что делает её оптимизацию сложной задачей.

Практическая значимость: Подобные функции часто встречаются в задачах калибровки моделей и в приложениях машинного обучения.

Широкое использование: Она является стандартом в сравнительных исследованиях методов оптимизации, что позволяет объективно оценить эффективность алгоритма CMA-ES.

Описание алгоритма CMA-ES

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) — это метод оптимизации, который адаптивно изменяет ковариационную матрицу распределения, используемую для генерации новых решений. Основные шаги алгоритма:

Инициализация: Начальная популяция генерируется из многомерного нормального распределения.

Оценка: Каждое решение оценивается с использованием целевой функции.

Выбор: Отбираются лучшие решения на основе их значений функции.

Обновление: Ковариационная матрица обновляется с учётом выбранных решений для направления поиска в перспективные области пространства решений.

2.2.2. Реализация и экспериментирование с NSGA-II

Задача многокритериальной оптимизации DTLZ2

Задача DTLZ2 (Deb-Thiele-Laumanns-Zitzler 2) — это одна из стандартных тестовых задач для многокритериальной оптимизации. Она имеет несколько целей, что делает её идеальной для тестирования и сравнения алгоритмов многокритериальной оптимизации.

$$f_1(x) = (1 + g(x_M)) \cos(x_1\pi/2) \cos(x_2\pi/2) \cdots \cos(x_{M-2}\pi/2) \cos(x_{M-1}\pi/2)$$
$$f_2(x) = (1 + g(x_M)) \cos(x_1\pi/2) \cos(x_2\pi/2) \cdots \cos(x_{M-2}\pi/2) \sin(x_{M-1}\pi/2)$$
$$\vdots$$
$$f_M(x) = (1 + g(x_M)) \sin(x_1\pi/2)$$

где $g(x_M) = \sum_{i=M}^{n} (x_i - 0.5)^2$.

Рисунок 2 - Формулировка задачи DTLZ2

Выбор задачи DTLZ2 обоснован следующими причинами:

Комплексность и масштабируемость: DTLZ2 позволяет изменять количество целей и размерность задачи, что полезно для тестирования масштабируемости алгоритмов.

Практическая значимость: Подобные многокритериальные задачи встречаются в реальном мире, например, в проектировании инженерных систем, управлении ресурсами и принятии решений в бизнесе.

Стандартизированные результаты: Задача DTLZ2 используется в академической литературе, что позволяет легко сравнивать результаты с другими исследованиями.

Описание алгоритма NSGA-II

NSGA-II (Non-dominated Sorting Genetic Algorithm II) — это усовершенствованная версия алгоритма NSGA, который широко используется для многокритериальной оптимизации. Основные компоненты NSGA-II:

Недоминирующая сортировка: Решения сортируются на основании их доминирования. Решения, не доминируемые другими, помещаются в первый фронт, и так далее.

Расстояние толерантности к толпе: Рассчитывается для каждого решения, чтобы сохранить разнообразие в популяции.

Операторы генетической алгоритмики: Включают кроссинговер, мутацию и селекцию, направленные на создание и улучшение решений.

NSGA-II применяется в задачах, где необходимо найти сбалансированные компромиссы между несколькими противоречивыми целями, например:

Оптимизация портфеля инвестиций, где цели — максимизация доходности и минимизация риска.

Проектирование автомобильных двигателей с целью минимизации расхода топлива и выбросов.

Эти задачи демонстрируют широкий спектр применения алгоритмов CMA-ES и NSGA-II в различных областях, что подчёркивает их важность и необходимость глубокого понимания и исследования.

2.3. Проведение экспериментов и анализ результатов

Для проведения экспериментов были выбраны следующие задачи оптимизации:

- Задача оптимизации функции Розенброка с использованием CMA-ES.
- Задача многокритериальной оптимизации DTLZ2 с использованием NSGA-II.

2.3.1. Оптимизация функции Розенброка с использованием CMA-ES

Функция Розенброка часто используется для тестирования алгоритмов оптимизации из-за наличия сложного ландшафта с узкой параболической долиной. Проведем эксперимент, сравнив начальные и конечные значения функции, количество итераций и время выполнения.

Результаты эксперимента:

Поколение	Среднее значение	Минимальное значение	Максимальное значение
0	123986	21214,6	323407
20	88,5	41,25	302,73
40	9,31	8,46	109,4
60	8,1	8,06	8,34
80	7,99	7,98	8
100	7,89	7,87	7,92

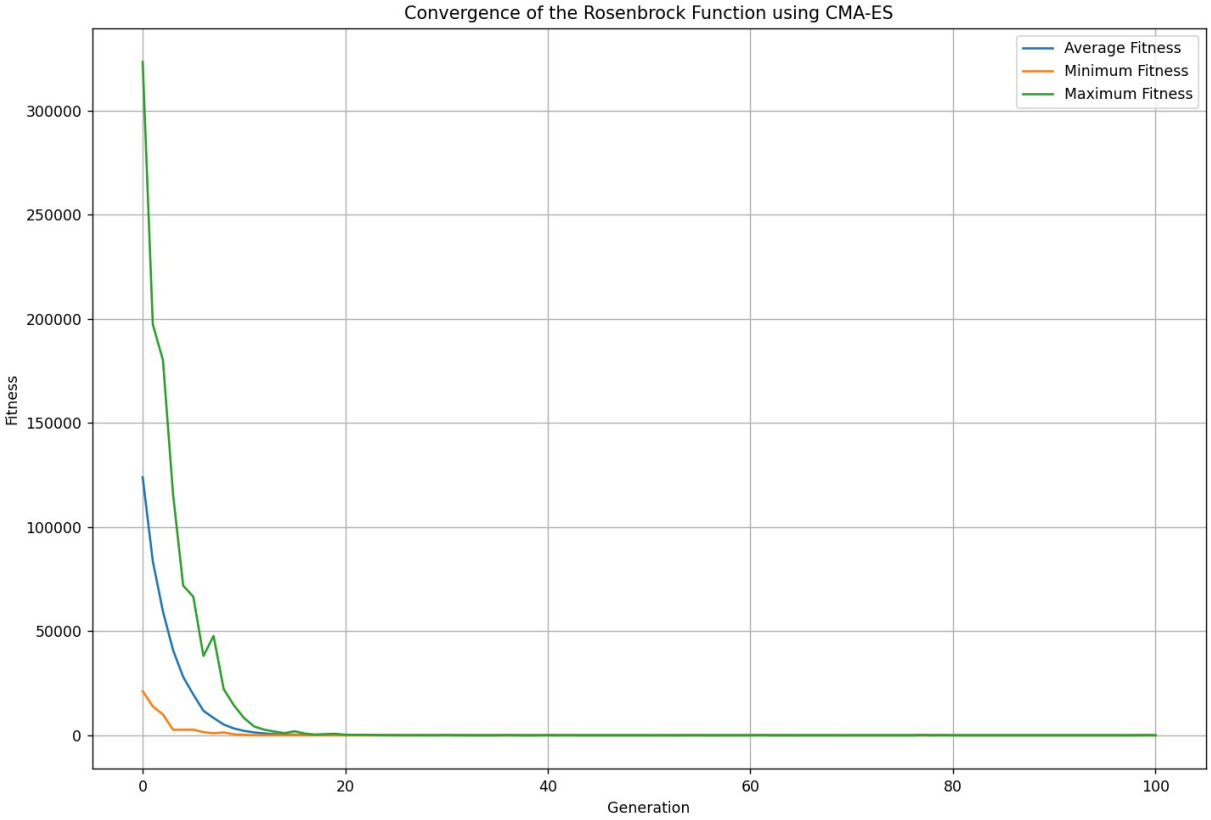

Рисунок 3 - Конвергенция функции Розенброка при использовании CMA-ES.

2.3.2. Многокритериальная оптимизация задачи DTLZ2 с использованием NSGA-II

Функция DTLZ2 является одной из стандартных задач многокритериальной оптимизации. Проведем эксперимент, сравнив результаты оптимизации, такие как количество недоминируемых решений и равномерность покрытия фронта Парето.

Результаты эксперимента:

Поколение	Количество не доминируемых решений	Среднее расстояние до фронта Парето	Количество вычислений
0	41	0,54	100
50	100	0.049	5000
100	100	0.046	10000
150	100	0.047	15000
200	100	0.048	20000

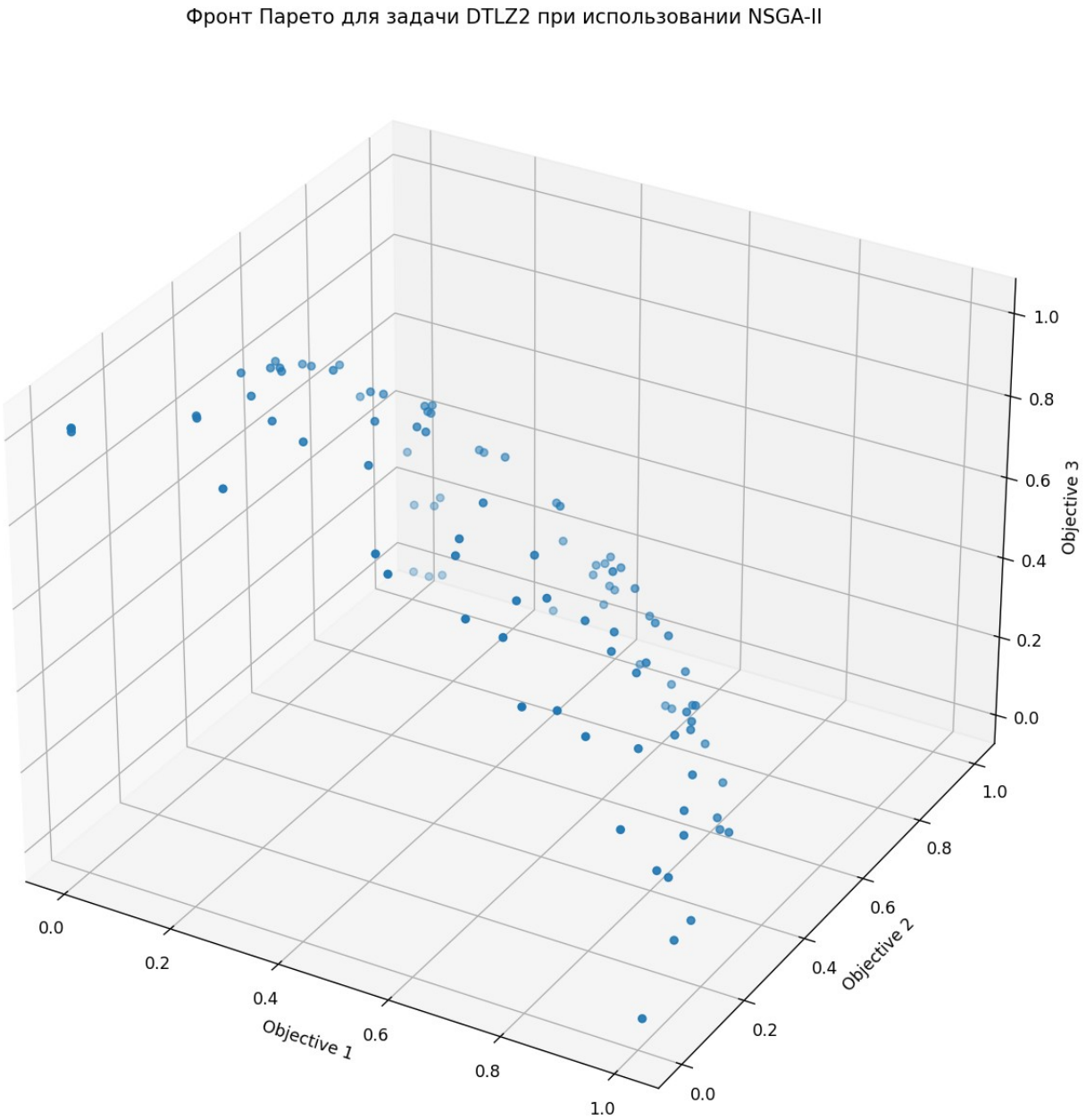

Рисунок 4 - фронт Парето для задачи DTLZ2 при использовании NSGA-II.

2.4. Обсуждение результатов и сравнительный анализ

В этом разделе обсуждаются результаты экспериментов, приводятся таблицы, графики и диаграммы для наглядного иллюстрирования полученных данных. Оценивается производительность каждого алгоритма, его эффективность в решении поставленных задач и делаются выводы о предпочтительности

использования того или иного алгоритма в зависимости от задачи.

2.4.1. Сравнительный анализ эффективности CMA-ES и NSGA

Параметр	CMA-ES	NSGA-II
Эффективность	Высокая для задач с непрерывными функциями	Высокая для многокритериальных задач
Сложность настройки	Низкая	Средняя
Области применения	Машинное обучение, оптимизация параметров	Инженерное проектирование, управление ресурсами
Время выполнения	Умеренное	Зависит от сложности задачи

2.5. Выводы

На основании проведённого анализа можно сделать следующие выводы:

1. Алгоритм CMA-ES показал высокую эффективность в задачах с непрерывными функциями, таких как функция Розенброка. Он продемонстрировал быструю сходимость и простоту настройки параметров.

2. Алгоритм NSGA-II оказался предпочтительным для задач многокритериальной оптимизации, таких как DTLZ2. Он обеспечил хорошее покрытие фронта Парето и высокую равномерность решений.

3. Выбор алгоритма зависит от конкретной задачи. Для задач с одной целью и непрерывными переменными рекомендуется использовать CMA-ES, тогда как для многокритериальных задач NSGA-II будет более эффективным.

4. Время выполнения алгоритмов варьируется в зависимости от сложности задачи и количества критериев, что также следует учитывать при выборе метода оптимизации.

Заключение

В заключении курсовой работы по прикладным моделям оптимизации, сосредоточенной на анализе эволюционных алгоритмов CMA-ES и NSGA-I/II/III, были рассмотрены как теоретические, так и практические аспекты этих методов оптимизации. Ключевые моменты исследования:

1. Теоретический анализ и практическая реализация: Были подробно изучены теоретические основы и принципы работы эволюционных алгоритмов CMA-ES и NSGA-I/II/III. Этот анализ позволяет глубже понять особенности каждого алгоритма и их применимость в различных областях.

2. Сравнительный анализ алгоритмов: проведён сравнительный анализ CMA-ES и NSGA-I/II/III, выявлены их преимущества, недостатки и области применения. Этот анализ позволяет лучше понять, в каких сценариях каждый из этих алгоритмов может быть наиболее эффективным.

3. Практическая значимость исследования: реализация и анализ этих алгоритмов на практике помогают в оценке их производительность и эффективность в решении реальных задач оптимизации. Были приведены примеры использования алгоритмов на различных наборах данных и задачах оптимизации.

4. Полученные результаты: В ходе исследования мы получили ряд интересных результатов, включая анализ эффективности алгоритмов на различных наборах данных, сравнение времени выполнения и качества найденных решений. Эти результаты могут быть полезны для исследователей и практиков в области оптимизации.

5. Перспективы дальнейших исследований: На основе полученных результатов можно выделить несколько направлений для дальнейших исследований, таких как улучшение алгоритмов оптимизации, исследование их применения в новых областях и разработка новых методов оценки эффективности алгоритмов.

Полученные результаты могут быть использованы как исследователями в

этой области, так и специалистами, работающими над конкретными практическими задачами оптимизации.

Список использованных источников.

- Саймон Д. Алгоритмы эволюционной оптимизации. — М.: ДМК Пресс, 2020. — 940 с. — ISBN 978-5-97060-812-8.
- Емельянов В. В., Курейчик В. В., Курейчик В. М. Теория и практика эволюционного моделирования. — М.: Физматлит, 2003. — 432 с. — ISBN 5-9221-0337-7.
- Курейчик В. М., Лебедев Б. К., Лебедев О. К. Поисковая адаптация: теория и практика. — М.: Физматлит, 2006. — 272 с. — ISBN 5-9221-0749-6.
- Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы: Учебное пособие. — 2-е изд. — М.: Физматлит, 2006. — 320 с. — ISBN 5-9221-0510-8.
- Гладков Л. А., Курейчик В. В, Курейчик В. М. и др. Биоинспирированные методы в оптимизации: монография. — М.: Физматлит, 2009. — 384 с. — ISBN 978-5-9221-1101-0.
- Оже, А.; Н. Хансен (2005). "Стратегия возобновления Эволюции CMA С увеличением Численности популяции".

Приложение 1

- Исходный код для реализации алгоритма CMA-ES

```
import numpy as np
import matplotlib.pyplot as plt
from deap import base, creator, tools, algorithms

# Определение задачи оптимизации
def rosenbrock(x):
    return sum(100.0 * (x[i+1] - x[i]**2.0)**2.0 + (1 - x[i])**2.0 for i in
range(len(x)-1)),

# Создание классов для DEAP
creator.create("FitnessMin", base.Fitness, weights=(-1.0,))
creator.create("Individual", np.ndarray, fitness=creator.FitnessMin)

# Определение инструментов DEAP
toolbox = base.Toolbox()
toolbox.register("attr_float", np.random.uniform, -5, 5)
toolbox.register("individual", tools.initRepeat, creator.Individual,
toolbox.attr_float, n=10)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
toolbox.register("mate", tools.cxBlend, alpha=0.5)
toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=1, indpb=0.2)
toolbox.register("select", tools.selTournament, tournsize=3)
toolbox.register("evaluate", rosenbrock)

# Кастомный метод сравнения для HallOfFame
def custom_similar(ind1, ind2):
    return np.array_equal(ind1, ind2)
```

```
# Основная функция для выполнения оптимизации
def main():
    np.random.seed(42)
    pop = toolbox.population(n=300)
    hof = tools.HallOfFame(1, similar=custom_similar)
    stats = tools.Statistics(lambda ind: ind.fitness.values)
    stats.register("avg", np.mean)
    stats.register("std", np.std)
    stats.register("min", np.min)
    stats.register("max", np.max)

    pop, log = algorithms.eaMuPlusLambda(pop, toolbox, mu=300, lambda_=300,
cxpb=0.7, mutpb=0.2, ngen=100, stats=stats, halloffame=hof, verbose=True)

    return pop, log, hof

if __name__ == "__main__":
    pop, log, hof = main()
    print("Best individual is ", hof[0], hof[0].fitness.values)

    # Построение графика конвергенции
    gen = log.select("gen")
    avg = log.select("avg")
    min_ = log.select("min")
    max_ = log.select("max")

    plt.figure(figsize=(10, 6))
    plt.plot(gen, avg, label="Average Fitness")
    plt.plot(gen, min_, label="Minimum Fitness")
```

```
plt.plot(gen, max_, label="Maximum Fitness")
plt.xlabel("Generation")
plt.ylabel("Fitness")
plt.title("Convergence of the Rosenbrock Function using CMA-ES")
plt.legend()
plt.grid()
plt.savefig('rosenbrock_convergence.png')
plt.show()
```

- Исходный код для реализации алгоритма NSGA

```
import numpy as np
import matplotlib.pyplot as plt
from pymoo.algorithms.moo.nsga2 import NSGA2
from pymoo.factory import get_problem
from pymoo.optimize import minimize

# Определение задачи многокритериальной оптимизации
problem = get_problem("dtlz2")

algorithm = NSGA2(pop_size=100)

res = minimize(problem,
               algorithm,
               ('n_gen', 200),
               verbose=True)

print("Best solutions found:")
print(res.F)
```

```
# Построение фронта Парето
fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(res.F[:, 0], res.F[:, 1], res.F[:, 2])

ax.set_xlabel('Objective 1')
ax.set_ylabel('Objective 2')
ax.set_zlabel('Objective 3')
ax.set_title('Фронт Парето для задачи DTLZ2 при использовании NSGA-II')
plt.savefig('pareto_front.png')
plt.show()
```

Приложение 2

Результат работы программ

1)

gen	nevals	avg	std	min	max
0	300	123986	54228	21214.6	323407
1	271	81472.6	35066.8	5438.65	197139
2	260	52493.8	23927.8	5438.65	155714
3	280	35921.9	18554.1	3977.45	122128
4	265	22157.9	12733.2	2330.1	78120.6
5	266	13359.6	6483.35	2330.1	57402.5
6	266	8569.53	4026.86	294.709	20068.2
7	270	5673.99	2830.2	294.709	16516.3
8	273	3580.17	1913.08	294.709	12049.3
9	263	2393.7	1440.1	294.709	15216.6
10	269	1481.31	756.488	294.709	5135.24
11	262	982.05	528.995	272.485	2875.99
12	253	644.895	375.775	182.157	2746.02
13	268	428.818	218.879	96.3366	1569.09
14	275	295.17	132.631	73.5478	1112.94
15	268	223.368	96.9737	34.8587	926.954
16	271	168.859	71.0436	26.0015	536.686
17	276	126.56	53.4056	26.0015	497.388
18	275	100.402	37.8123	26.0015	324.626
19	272	76.4248	26.3231	26.0015	185.627
20	270	58.953	21.1989	21.1243	266.165
21	274	46.8238	15.3051	18.4916	123.774
22	277	38.1005	15.19	18.1596	222.694
23	264	30.1493	7.41468	17.5978	58.6224
24	266	24.7127	5.35363	13.0979	49.2283
25	272	20.8199	4.6037	11.3614	40.1803
26	266	17.3342	3.67195	10.1471	30.776
27	271	14.8275	3.20148	8.54048	31.4836
28	274	12.8938	2.12208	8.54048	21.3851
29	269	11.6965	4.23386	7.60739	78.5998
30	277	10.4015	1.73265	7.26796	19.7428

31	267	9.48881	1.22706	7.26796	17.7103
32	274	8.8132	1.14501	6.62054	15.8423
33	281	8.10963	0.909293	6.02076	12.9979
34	260	7.69816	0.879824	6.02076	12.2759
35	270	7.29544	0.749538	6.02076	12.522
36	262	6.98224	0.527444	5.99754	9.15123
37	273	6.66753	0.472808	5.81991	8.86487
38	273	6.62297	4.70887	5.54201	87.7946
39	271	6.12467	0.231381	5.54201	6.96643
40	277	5.96418	0.188475	5.50937	6.70674
41	264	5.82215	0.144632	5.51987	6.24712
42	268	5.74047	0.124936	5.48991	6.36535
43	261	5.66096	0.0891568	5.48086	5.93769
44	268	5.59768	0.0810871	5.39887	6.0677
45	276	5.55417	0.0801892	5.39887	6.19182
46	272	5.5044	0.0543962	5.3455	5.66655
47	279	5.46701	0.0529068	5.3455	5.71106
48	272	5.43009	0.0448679	5.3336	5.60774
49	266	5.3975	0.034079	5.3336	5.619
50	269	5.37176	0.0233552	5.31035	5.45882
51	262	5.35688	0.0208136	5.29122	5.43259
52	281	5.34135	0.0180009	5.29122	5.42075
53	264	5.32955	0.0175894	5.29122	5.40665
54	263	5.3241	0.164725	5.26986	8.159
55	255	5.30286	0.0147698	5.26986	5.33992
56	272	5.29001	0.0148856	5.20945	5.33992
57	280	5.28005	0.0146131	5.20945	5.34018
58	269	6.38733	19.3151	5.20945	340.376
59	261	5.25999	0.0126044	5.20945	5.30698
60	261	5.2517	0.0159304	5.20667	5.38573
61	269	5.8468	10.4568	5.20667	186.662
62	269	5.42067	3.23685	5.19983	61.3903
63	266	5.22425	0.0152757	5.19165	5.278
64	271	5.21466	0.0118431	5.19165	5.2549
65	276	5.20727	0.00900398	5.18996	5.2484

66	273	5.20176	0.00702347	5.17898	5.23932
67	271	5.19665	0.00610702	5.17898	5.21398
68	260	5.19159	0.00534731	5.17898	5.21225
69	264	5.41906	4.00301	5.17284	74.6374
70	268	5.1838	0.00447042	5.17284	5.20111
71	264	6.15712	16.8817	5.17284	298.068
72	272	5.17789	0.0032177	5.16121	5.18784
73	268	5.17563	0.0027825	5.16231	5.1878
74	269	5.17371	0.0025804	5.16231	5.18415
75	269	5.5608	6.72407	5.16231	121.831
76	278	5.16973	0.00282739	5.15822	5.17442
77	275	5.16746	0.00344219	5.15372	5.17359
78	272	5.16463	0.00353401	5.15372	5.17455
79	270	5.16232	0.00371285	5.15094	5.17607
80	271	5.15935	0.00386646	5.14986	5.18145
81	273	5.1572	0.00364361	5.14865	5.17768
82	270	5.1545	0.00288983	5.14663	5.16459
83	275	5.15191	0.00297552	5.13443	5.16003
84	269	5.3123	2.81831	5.13443	54.0454
85	275	5.20271	0.967793	5.13443	21.9373
86	267	5.14409	0.00375232	5.13443	5.15095
87	258	5.46453	5.59683	5.13338	102.243
88	271	5.13809	0.00288816	5.13338	5.14857
89	266	5.1362	0.0020907	5.12934	5.14616
90	273	5.13465	0.00170397	5.12934	5.14172
91	271	6.48625	23.3964	5.1274	411.048
92	264	5.1317	0.00187982	5.12446	5.13571
93	266	5.47555	5.97228	5.12368	108.746
94	271	5.12865	0.0019637	5.11883	5.13249
95	281	5.12702	0.00223343	5.11883	5.13321
96	273	5.12524	0.00218175	5.11883	5.13087
97	272	5.12358	0.00226784	5.11809	5.129
98	272	5.39063	4.64843	5.11671	85.7694
99	275	5.13341	0.226553	5.11343	9.05072
100	267	5.12395	0.0882536	5.11343	6.64974

Best individual is [8.89345349e-01 7.93604970e-01 6.37846892e-01 4.16809520e-01
1.82052753e-01 4.00748890e-02 8.79692805e-03 9.81811688e-03
1.13656627e-02 -5.59487276e-04] (5.113178070451651,)

2)

n_gen	n_eval	n_nds	igd	gd
1	100	43	0.4329898768	0.5625459471
2	200	54	0.4153954218	0.5332666059
3	300	66	0.3600033325	0.5223164219
4	400	65	0.3398101680	0.4685325167
5	500	70	0.3100019649	0.4354131291
6	600	71	0.2761302653	0.4111826985
7	700	77	0.2552593395	0.3652783591
8	800	83	0.2389245422	0.3374281117
9	900	100	0.2202036761	0.3264885360
10	1000	90	0.1877912984	0.2649291255
11	1100	100	0.1751588389	0.2411797029
12	1200	100	0.1616453930	0.2032632690
13	1300	100	0.1493307101	0.1656733708
14	1400	100	0.1387745933	0.1474539540
15	1500	100	0.1307274080	0.1325890763
16	1600	100	0.1241576747	0.1217834730
17	1700	100	0.1217621856	0.1166389968
18	1800	100	0.1236949621	0.1161749598
19	1900	100	0.1179626555	0.1109081568
20	2000	100	0.1116985346	0.1082091245
21	2100	100	0.1020526493	0.1002094838
22	2200	100	0.0993310904	0.0929175438
23	2300	100	0.0970275550	0.0906468264
24	2400	100	0.0975727307	0.0852040874
25	2500	100	0.0930257594	0.0830618018
26	2600	100	0.0947401242	0.0883562153
27	2700	100	0.0943004533	0.0880037776

28 | 2800 | 100 | 0.0978570911 | 0.0916024260
29 | 2900 | 100 | 0.0969603185 | 0.0893497797
30 | 3000 | 100 | 0.0954609168 | 0.0847982729
31 | 3100 | 100 | 0.0952931752 | 0.0842498476
32 | 3200 | 100 | 0.0941536028 | 0.0781583681
33 | 3300 | 100 | 0.0914393458 | 0.0780573860
34 | 3400 | 100 | 0.0887638763 | 0.0743220193
35 | 3500 | 100 | 0.0856710910 | 0.0729876743
36 | 3600 | 100 | 0.0853812126 | 0.0724059144
37 | 3700 | 100 | 0.0871762617 | 0.0716559243
38 | 3800 | 100 | 0.0834689009 | 0.0741963497
39 | 3900 | 100 | 0.0821293865 | 0.0681566090
40 | 4000 | 100 | 0.0848305595 | 0.0677340417
41 | 4100 | 100 | 0.0843339600 | 0.0652389777
42 | 4200 | 100 | 0.0832408000 | 0.0649860101
43 | 4300 | 100 | 0.0866116820 | 0.0602557796
44 | 4400 | 100 | 0.0832165174 | 0.0586290553
45 | 4500 | 100 | 0.0788454168 | 0.0556268567
46 | 4600 | 100 | 0.0786016949 | 0.0548641306
47 | 4700 | 100 | 0.0784424970 | 0.0561557884
48 | 4800 | 100 | 0.0779918882 | 0.0560431649
49 | 4900 | 100 | 0.0756118603 | 0.0543089929
50 | 5000 | 100 | 0.0738558659 | 0.0537782573
51 | 5100 | 100 | 0.0839826494 | 0.0544880377
52 | 5200 | 100 | 0.0774487350 | 0.0545417323
53 | 5300 | 100 | 0.0779336827 | 0.0530190287
54 | 5400 | 100 | 0.0766783196 | 0.0528965636
55 | 5500 | 100 | 0.0766303081 | 0.0510968319
56 | 5600 | 100 | 0.0775609760 | 0.0536067921
57 | 5700 | 100 | 0.0799204291 | 0.0543006438
58 | 5800 | 100 | 0.0815779536 | 0.0520238513
59 | 5900 | 100 | 0.0812153801 | 0.0494210144
60 | 6000 | 100 | 0.0758422978 | 0.0488781115
61 | 6100 | 100 | 0.0743974719 | 0.0511250455
62 | 6200 | 100 | 0.0760290041 | 0.0510964306

63	6300	100	0.0770935852	0.0511572460
64	6400	100	0.0799882506	0.0503372721
65	6500	100	0.0801295015	0.0492042432
66	6600	100	0.0758153593	0.0506872768
67	6700	100	0.0774971957	0.0499492320
68	6800	100	0.0753430276	0.0484703179
69	6900	100	0.0749560137	0.0501793554
70	7000	100	0.0757521421	0.0503221170
71	7100	100	0.0740699633	0.0499294034
72	7200	100	0.0753943948	0.0478381165
73	7300	100	0.0772922987	0.0484801769
74	7400	100	0.0793012023	0.0489587554
75	7500	100	0.0787924593	0.0491830384
76	7600	100	0.0804736778	0.0489609786
77	7700	100	0.0831729753	0.0481605865
78	7800	100	0.0765874407	0.0473506877
79	7900	100	0.0753450192	0.0465191019
80	8000	100	0.0751037874	0.0477052292
81	8100	100	0.0735693093	0.0476407430
82	8200	100	0.0697136023	0.0479931713
83	8300	100	0.0728613468	0.0478667630
84	8400	100	0.0707819805	0.0465029938
85	8500	100	0.0703211159	0.0469741105
86	8600	100	0.0713844552	0.0458414477
87	8700	100	0.0719711152	0.0445923770
88	8800	100	0.0716980349	0.0465520890
89	8900	100	0.0722064098	0.0479918739
90	9000	100	0.0748808132	0.0479264496
91	9100	100	0.0764197255	0.0486352671
92	9200	100	0.0748705696	0.0475158624
93	9300	100	0.0771829882	0.0509813652
94	9400	100	0.0769017393	0.0514435335
95	9500	100	0.0694860597	0.0484621476
96	9600	100	0.0770516158	0.0497855379
97	9700	100	0.0711788084	0.0478147380

98	9800	100	0.0730428722	0.0486729219
99	9900	100	0.0746483615	0.0493914768
100	10000	100	0.0771182355	0.0492282303
101	10100	100	0.0713906062	0.0494447538
102	10200	100	0.0724843439	0.0501224869
103	10300	100	0.0711006904	0.0490569709
104	10400	100	0.0711646899	0.0473306420
105	10500	100	0.0725059287	0.0476268033
106	10600	100	0.0733462295	0.0486014596
107	10700	100	0.0708102247	0.0462345234
108	10800	100	0.0704343824	0.0442357295
109	10900	100	0.0713921770	0.0465686110
110	11000	100	0.0702250220	0.0457873546
111	11100	100	0.0706594195	0.0460980233
112	11200	100	0.0691911331	0.0467229788
113	11300	100	0.0721080640	0.0463858224
114	11400	100	0.0705732365	0.0465246632
115	11500	100	0.0741670181	0.0455453315
116	11600	100	0.0717312567	0.0449827624
117	11700	100	0.0714553488	0.0449642349
118	11800	100	0.0693222074	0.0454055637
119	11900	100	0.0730187563	0.0457685641
120	12000	100	0.0725266569	0.0443221780
121	12100	100	0.0673188118	0.0419615504
122	12200	100	0.0773580480	0.0454959088
123	12300	100	0.0752869688	0.0441759769
124	12400	100	0.0733902256	0.0439488375
125	12500	100	0.0782525911	0.0432693307
126	12600	100	0.0754101311	0.0457092989
127	12700	100	0.0777604701	0.0479102066
128	12800	100	0.0749550521	0.0446478383
129	12900	100	0.0781923227	0.0476712842
130	13000	100	0.0714525263	0.0477395517
131	13100	100	0.0723980032	0.0488866418
132	13200	100	0.0693678450	0.0482339532

133 | 13300 | 100 | 0.0731610427 | 0.0467324119
134 | 13400 | 100 | 0.0748351061 | 0.0483479975
135 | 13500 | 100 | 0.0731790249 | 0.0483226027
136 | 13600 | 100 | 0.0691257755 | 0.0473020063
137 | 13700 | 100 | 0.0695313103 | 0.0470566764
138 | 13800 | 100 | 0.0650579407 | 0.0478310374
139 | 13900 | 100 | 0.0696617703 | 0.0457106624
140 | 14000 | 100 | 0.0703180121 | 0.0440090222
141 | 14100 | 100 | 0.0714146616 | 0.0443046466
142 | 14200 | 100 | 0.0690921427 | 0.0438328667
143 | 14300 | 100 | 0.0681199070 | 0.0443671129
144 | 14400 | 100 | 0.0702926662 | 0.0456531095
145 | 14500 | 100 | 0.0710497436 | 0.0447854632
146 | 14600 | 100 | 0.0734999513 | 0.0439531371
147 | 14700 | 100 | 0.0740557250 | 0.0427658176
148 | 14800 | 100 | 0.0727614647 | 0.0448412435
149 | 14900 | 100 | 0.0700505822 | 0.0462309761
150 | 15000 | 100 | 0.0714128300 | 0.0451688504
151 | 15100 | 100 | 0.0706797663 | 0.0458631765
152 | 15200 | 100 | 0.0728708384 | 0.0457150330
153 | 15300 | 100 | 0.0751266126 | 0.0474273749
154 | 15400 | 100 | 0.0732217667 | 0.0475338969
155 | 15500 | 100 | 0.0694863811 | 0.0468925628
156 | 15600 | 100 | 0.0727531183 | 0.0463226760
157 | 15700 | 100 | 0.0725319332 | 0.0465453141
158 | 15800 | 100 | 0.0726997640 | 0.0487303595
159 | 15900 | 100 | 0.0745144268 | 0.0490592283
160 | 16000 | 100 | 0.0777882385 | 0.0491567144
161 | 16100 | 100 | 0.0748468113 | 0.0484976036
162 | 16200 | 100 | 0.0676218398 | 0.0466847201
163 | 16300 | 100 | 0.0699742844 | 0.0464155098
164 | 16400 | 100 | 0.0715683533 | 0.0469128467
165 | 16500 | 100 | 0.0684130584 | 0.0450025237
166 | 16600 | 100 | 0.0705940892 | 0.0454160642
167 | 16700 | 100 | 0.0700174337 | 0.0458458918

168	16800	100	0.0680758192	0.0452607263
169	16900	100	0.0683480572	0.0450890218
170	17000	100	0.0723664583	0.0461627460
171	17100	100	0.0718333826	0.0469656472
172	17200	100	0.0675390428	0.0458718513
173	17300	100	0.0689890674	0.0460470229
174	17400	100	0.0690612059	0.0482120681
175	17500	100	0.0673000452	0.0451659800
176	17600	100	0.0679898673	0.0453874491
177	17700	100	0.0687906084	0.0462748662
178	17800	100	0.0678652575	0.0461670429
179	17900	100	0.0674161366	0.0447260065
180	18000	100	0.0694915727	0.0452930584
181	18100	100	0.0716671848	0.0467198196
182	18200	100	0.0720904441	0.0471071811
183	18300	100	0.0740943907	0.0470485240
184	18400	100	0.0733904112	0.0472646013
185	18500	100	0.0721770393	0.0451064655
186	18600	100	0.0719934891	0.0452458874
187	18700	100	0.0685890788	0.0449329892
188	18800	100	0.0715480615	0.0442267450
189	18900	100	0.0717214507	0.0460060142
190	19000	100	0.0720269478	0.0465078997
191	19100	100	0.0701288538	0.0467887915
192	19200	100	0.0703610517	0.0478830813
193	19300	100	0.0690915951	0.0478035607
194	19400	100	0.0667885894	0.0468013256
195	19500	100	0.0702526262	0.0470164356
196	19600	100	0.0686411880	0.0463522991
197	19700	100	0.0724662116	0.0458562138
198	19800	100	0.0699734588	0.0450754816
199	19900	100	0.0673727828	0.0440088737
200	20000	100	0.0714412241	0.0449027363

Best solutions found:

[[2.36369312e-03 1.01159194e+00 4.35393378e-02]

[4.52745685e-07 7.52270942e-07 1.00852730e+00]
[1.00349803e+00 1.15804766e-04 2.35605090e-02]
[8.86261063e-01 4.68718886e-01 1.07539036e-12]
[4.38918454e-12 2.58362635e-06 1.00691788e+00]
[4.29298977e-06 1.04467154e-14 1.00106544e+00]
[7.66916274e-01 5.63018775e-01 3.41529429e-01]
[4.54541508e-01 7.54549668e-01 4.90618539e-01]
[9.60927227e-01 2.89137882e-01 9.86357368e-02]
[8.70535262e-01 4.38914051e-01 3.11916939e-01]
[1.45194733e-01 8.50903693e-01 5.34974152e-01]
[8.11245115e-02 4.97139058e-01 8.94178995e-01]
[4.94633276e-01 8.62892071e-01 1.54765967e-01]
[7.13034146e-01 7.01353414e-01 6.65326063e-02]
[1.93998631e-01 9.37449978e-01 3.00718590e-01]
[9.63143631e-01 1.80650934e-01 2.31323462e-01]
[6.81625606e-01 6.39073704e-01 3.66177850e-01]
[4.07631278e-01 7.40993221e-02 9.15956257e-01]
[1.57077624e-01 9.62587272e-01 2.70306170e-01]
[6.94738766e-01 2.29094018e-01 6.85502661e-01]
[1.83465664e-01 9.56981662e-01 2.45811785e-01]
[3.74944315e-01 8.85307538e-01 2.84081098e-01]
[3.84224282e-01 9.07219126e-01 1.93737857e-01]
[2.25867850e-01 6.78532712e-01 7.20174367e-01]
[4.66091026e-01 7.71148568e-01 4.55120895e-01]
[4.42600904e-01 6.61585705e-01 6.11304295e-01]
[5.62237774e-01 8.08671461e-01 1.85201686e-01]
[9.96567406e-02 9.83466257e-01 2.10496871e-01]
[5.44122929e-01 3.08824660e-01 7.85124240e-01]
[3.70138082e-02 8.33016922e-01 5.68047312e-01]
[8.95665140e-01 4.52291459e-01 8.47664341e-02]
[5.78564201e-01 8.20731589e-01 2.46415148e-02]
[4.44098430e-02 9.99468914e-01 3.41297841e-02]
[2.11315855e-01 3.52841181e-01 9.12990059e-01]
[2.92666646e-01 7.22359096e-01 6.37281217e-01]
[3.35570533e-01 1.10656272e-01 9.62615844e-01]

[9.73732919e-01 1.52508485e-01 2.09879567e-01]
[3.13867845e-01 7.08370507e-01 6.36362561e-01]
[7.81157688e-01 5.49145482e-01 3.26090000e-01]
[9.91465527e-01 9.52248959e-02 1.25766894e-01]
[2.25866878e-01 4.29482731e-01 9.08879016e-01]
[1.04398385e-01 9.84505193e-01 1.69735072e-01]
[7.03268333e-01 3.74387935e-01 6.10033241e-01]
[3.11647582e-01 5.66513803e-02 9.70103954e-01]
[4.28067665e-01 6.21144265e-01 6.59332483e-01]
[5.69658940e-02 7.04079216e-01 7.10016814e-01]
[8.45801624e-01 2.47191210e-02 5.61571494e-01]
[6.45744689e-01 1.52666149e-01 7.58389296e-01]
[3.18140034e-02 7.15992340e-01 6.99532149e-01]
[2.41598968e-01 6.55185100e-01 7.32674352e-01]
[6.95841882e-02 1.84618277e-01 9.82211444e-01]
[6.61824402e-01 3.37601874e-01 6.72904021e-01]
[6.72576693e-01 7.09644300e-01 2.49388005e-01]
[8.16361055e-01 1.69487052e-01 5.67999932e-01]
[7.47001501e-01 6.16056374e-02 6.64577621e-01]
[1.38841741e-01 3.90620446e-01 9.11254904e-01]
[6.31768958e-01 5.94053008e-01 5.19064337e-01]
[2.51703850e-01 2.47580506e-01 9.40308055e-01]
[5.34713690e-01 2.87068821e-01 8.03006728e-01]
[7.29563156e-01 2.61636626e-01 6.39773421e-01]
[9.62990130e-01 2.09246851e-01 2.93630400e-01]
[9.13596545e-01 1.03501150e-01 4.14546897e-01]
[6.87117357e-01 3.94116717e-02 7.27297711e-01]
[3.86489557e-01 3.43964163e-02 9.36684676e-01]
[1.20523810e-01 7.38581868e-01 6.70231882e-01]
[3.51412899e-01 5.83372979e-01 7.39145655e-01]
[5.28523000e-01 1.88122510e-01 8.29225873e-01]
[3.40959792e-01 7.69514508e-01 5.51248836e-01]
[2.85385741e-02 7.76168141e-02 9.97575734e-01]
[1.03251427e-01 9.73689072e-01 2.08338026e-01]
[6.20138753e-01 4.14494445e-01 6.77699090e-01]

[7.89688408e-01 6.09290571e-01 1.45818305e-01]
[9.48457952e-01 2.10545625e-02 3.20159692e-01]
[2.76196423e-01 3.91039377e-01 8.86945021e-01]
[2.43556219e-01 4.84137142e-01 8.77964242e-01]
[7.82637497e-01 6.20902369e-01 1.30855651e-01]
[8.55552190e-01 4.75863747e-01 2.78861675e-01]
[6.74352379e-01 2.21885671e-01 7.14917786e-01]
[9.93820612e-01 1.23756473e-01 2.12482866e-02]
[1.75086550e-01 9.41988318e-01 2.99534542e-01]
[4.75714696e-01 8.52500518e-01 2.90885365e-01]
[2.63848766e-01 5.02151228e-01 8.27997500e-01]
[8.54682616e-01 1.55496793e-01 4.95729745e-01]
[9.71959887e-01 2.76267286e-01 7.95194135e-02]
[7.63588486e-01 5.33804209e-01 3.77496165e-01]
[3.57459176e-01 7.82968751e-01 5.15179803e-01]
[5.25702057e-01 2.68164787e-01 8.10245194e-01]
[3.43124768e-01 1.25003479e-02 9.40523629e-01]
[9.06807942e-01 1.62395379e-01 4.23383912e-01]
[1.65088011e-01 1.08455962e-01 9.83357352e-01]
[3.30260371e-01 5.92056567e-01 7.35781701e-01]
[2.49990565e-01 4.05148266e-01 8.81985819e-01]
[2.21458107e-02 5.40991585e-01 8.56553204e-01]
[1.67070813e-02 1.00591446e+00 4.37199754e-02]
[5.89044960e-01 7.35785165e-01 3.76231932e-01]
[7.60528587e-01 2.61658465e-01 5.95891316e-01]
[5.70812967e-01 7.96863125e-01 2.92066680e-01]
[5.49405112e-01 7.44835527e-01 3.96187262e-01]
[9.20092563e-01 3.21051180e-01 2.71220837e-01]
[2.77987388e-01 4.82091011e-01 8.36569419e-01]]