МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра:**

**Информатики и автоматизации научных исследований**

Направление подготовки: «Прикладная информатика»

Профиль подготовки: «Прикладная информатика в области обработки данных»

**ОТЧЕТ**

по производственной практике

на тему:

**«Сравнительный анализ подходов к решению задачи об оптимальном распределении инвестиций»**

**Выполнил(а):** студент группы 381407в

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Иванеженков Д. В.

Подпись

**Научный руководитель:**

Старший преподаватель, кандидат технических наук

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Неймарк Е. А.

Подпись

Нижний Новгород  
2018

Оглавление

[Введение 3](#_Toc532964207)

[Формальная постановка задачи 4](#_Toc532964208)

[Обзор методов решения 5](#_Toc532964209)

[Алгоритм полного перебора 5](#_Toc532964210)

[Метод ветвей и границ 5](#_Toc532964211)

[Метод динамического программирования 5](#_Toc532964212)

[Жадные алгоритмы 6](#_Toc532964213)

[Генетические алгоритмы 6](#_Toc532964214)

[Алгоритмическое описание выбранных алгоритмов 8](#_Toc532964215)

[Метод динамического программирования 8](#_Toc532964216)

[Генетические алгоритмы 8](#_Toc532964217)

## Введение

Задача о ранце является широко известной задачей дискретной оптимизации, что обусловлено большим количеством её приложений.

Задача о ранце и её различные вариации находят широкое практическое применение в областях планирования и управления производственными и транспортными системами, прикладной математике, криптографии, и т.д.

Формулируется задача таким образом:

Пусть есть N разных предметов, каждый предмет имеет вес  и полезность , так же имеется максимальный вес , который можно положить в рюкзак. Требуется собрать такой набор предметов , чтобы полезность их была наибольшей, а суммарный вес не превышал .

В данной работе рассматривается задача оптимального распределения инвестиций, что является частным случаем задачи о ранце, а именно ранец с мультивыбором (англ. *Multiple-choice Knapsack Problem*) в приложении к экономике. В этой вариации предметы разделены на группы, и из каждой группы требуется выбрать только один предмет.

Задача является NP – трудной, т.е. на данный момент не существует алгоритмов, способных найти оптимальное решение за полиномиальное время. Отсюда вытекает проблема выбора метода для поиска решения: использовать точный алгоритм, дающий оптимальное решение, но не работающий за приемлемое время для больших объемов входных данных; либо использовать быстрый алгоритм, не гарантирующий нахождения оптимума.

Цель данной работы – провести сравнительный анализ методов решения задачи оптимального распределения инвестиций, выявить диапазоны наборов входных данных, на которых оправданно применение этих методов.

Для достижения поставленной цели необходимо решение следующих задач:

* Описание методов
* Реализация конкретных алгоритмов
* Проведение экспериментов с замерами времени работы и оценкой приближенности решения к оптимуму реализованных алгоритмов на различных наборах данных
* Сравнение и анализ собранных результатов
* Формирование заключения на основе полученных данных

## Формальная постановка задачи

Требуется вложить *T* имеющихся средств в *m* предприятий. Для каждого предприятия предусмотрены n сумм вложений. Прибыль , для каждого предприятия определяется в зависимости от количества вложенных средств , . Необходимо распределить средства так, чтобы прибыль со всех предприятий была максимальной. При этом, в одно предприятие можно вложиться только один раз.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | *…* |  | *…* |  |
|  |  |  | *…* |  | *…* |  |
|  |  |  | *…* |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  | *…* |  | *…* |  |
| *…* | *…* | *…* | *…* | *…* | *…* | *…* |
|  |  |  | *…* |  | *…* |  |

Здесь – прибыль *j*-го предприятия при вложении в него средств.

Математическая модель задачи:

Определить вектор , где , ∀ , где – количество средств, вложенных в *j*-е предприятие; удовлетворяющий условиям:

и обеспечивающий максимум целевой функции

## Обзор методов решения

# **Алгоритм полного перебора**

Полный перебор – метод решения, относящийся к классу методов поиска путем исчерпывания возможных вариантов. Любая задача из класса NP может быть решена полным перебором, но сложность вычислений зависит от количества всех решений задачи. В данном случае, для каждой группы из предметов (для каждого предприятия) существует n вариантов выбора предмета (вложения определенной суммы в предприятие). Тогда, если всего предприятий , перебор всех возможных вариантов имеет временную сложность *O*(), что позволяет использовать его лишь для небольшого количества предметов. С ростом числа предметов задача становится неразрешимой данным методом за приемлемое время.

# Метод ветвей и границ

Метод ветвей и границ – общий алгоритмический метод нахождения оптимальных решений различных задач оптимизации. Является вариацией метода полного перебора; отличается исключением заведомо неоптимальных ветвей дерева полного перебора на основе оценок верхних и нижних границ ожидаемого решения для каждой ветви.

В начале берется какое-либо допустимое решение, называемое «рекордом», для него считается значение целевой функции. Далее, в процессе построения дерева перебора, для каждого узла считается верхняя оценка возможного решения. Если эта оценка не больше значения целевой функции у «рекорда», то в рассматриваемом подмножестве не содержится решения лучше «рекорда», и оно может быть отброшено. Если значение целевой функции на очередном решении больше рекордного, то происходит смена рекорда. Алгоритм заканчивает свою работу, когда будут просмотрены все возможные подмножества. Метод ветвей и границ в среднем работает быстрее алгоритма полного перебора, но всегда можно подобрать такие входные данные, для которых оценка по времени работы этих алгоритмов будет совпадать.

# Метод динамического программирования

Динамическое программирование – подход, позволяющий решать задачи оптимизации, которые могут быть сформулированы как задачи многошагового оптимального управления некоторой системой. Метод основан на уравнении Беллмана и являет собой сведение сложной оптимизационной задачи к упорядоченной последовательности более простых задач (задач меньшей размерности) и их решения. При этом, на каждом шаге для решения задачи используются результаты решения предыдущих задач.

В общем случае, можно решить задачу, в которой присутствует оптимальная подструктура, проделав следующие шаги:

1. Разбиение задачи на подзадачи меньшего размера.
2. Нахождение оптимального решения подзадач рекурсивно, проделывая такой же трехшаговый алгоритм.
3. Использование полученного решения подзадач для конструирования решения исходной задачи.

Алгоритм является псевдополиномиальным, т.е. это алгоритм, проявляющий экспоненциальный характер только при относительно больших значениях числовых параметров.

# Жадные алгоритмы

Жадный алгоритм – метод решения оптимизационных задач, основанный на том, что процесс принятия решения можно разбить на элементарные шаги, на каждом из которых принимается отдельное решение. Решение, принимаемое на каждом шаге, оптимально только на текущем шаге и принимается без учета предыдущих или последующих решений; т.е. производится локально оптимальный выбор в надежде, что он приведет к оптимальному решению глобальной задачи.

Генетические алгоритмы

Генетический алгоритм – это эвристический алгоритм поиска, применяемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров, основываясь на моделировании биологических механизмов эволюции и популяционной генетике, а не на математических свойствах целевой функции. Генетические алгоритмы не гарантируют нахождения оптимального решения за полиномиальное время и не дают оценку близости решения к оптимальному, но обладают хорошими временными показателями, позволяя найти *достаточно хорошее* решение быстрее других известных детерминированных или эвристических методов.

Для нахождения решения при помощи генетического алгоритма, задача формализуется таким образом, чтобы её решение возможно было закодировать в виде вектора генов (генотипа). Некоторым образом, создается множество генотипов начальной популяции. Каждый генотип оценивается функцией приспособленности, в результате чего каждому генотипу ставится в соответствие определённое значение (приспособленность), определяющее насколько хорошо описываемый им фенотип решает поставленную задачу. Из полученного множества решений (поколения) путём применения генетических операторов скрещивания (кроссовера) и мутации строится множество новых решений (потомки). Для них также вычисляется значение функции приспособленности, и затем, производится отбор (селекция) решений в следующее поколение, основываясь на значении приспособленности.

Эти действия повторяются итеративно, моделируя эволюционный процесс, продолжающийся несколько циклов. Критерием остановки алгоритма могут быть: исчерпание числа поколений, исчерпание времени или нахождение субоптимального решения.

В качестве исследуемых алгоритмов, рассмотрим метод динамического программирования, несколько реализаций жадного и несколько вариаций генетических алгоритмов. Для метода динамического программирования можно приближенно оценить, начиная с какого объёма входных данных, он проявляет экспоненциальный характер. Также, этот алгоритм является точным, что позволяет оценить приближенность решений, получаемых в ходе работы генетических алгоритмов.

## Алгоритмическое описание выбранных алгоритмов

# Метод динамического программирования

Пусть есть предприятий. Для каждого предприятия есть возможных размеров вложений . Всего имеется средств. Прибыль от вложения средств в -е предприятие обозначим . Обозначим – количество средств, оставшихся после вложения в -е предприятие; – количество средств, вложенных в -е предприятие. Рассматриваем задачу с конца, т.е. допустим, что средства уже вложены в предприятие. Тогда, максимальная прибыль от вложения в -е предприятие находится по формуле: .

Отсюда, максимальная суммарная прибыль от вложения в и -е предприятия:

Аналогичным образом строим соотношения для оставшихся шагов. На последнем шаге получаем: . На каждом шаге запоминаем управление , при котором достигается максимум. Наконец, посчитав все соотношения, получаем – вектор решения, на котором достигается максимум целевой функции .

# Генетические алгоритмы

Генетические алгоритмы включают в себя множество различных операторов, работающих последовательно, и влияющих на формирование популяций решений. Все операторы, использующиеся в данной работе описаны ниже.

**Кодирование:**

Генотип представляет из себя вектор , где , ∀ .

- индекс размера инвестиций в -е предприятие ( – размер вложений в предприятие);

~ нулевое вложение;

– количество предложений;

– количество компаний;

**Формирование начальной популяции:**

Начальная популяция формируется одним из четырёх различных случайных операторов:

1. Каждому гену случайным образом присваивается значение в диапазоне от 0 до .
2. Каждому гену, начиная с первого случайным образом присваивается значение в диапазоне от 0 до , где – сумма всех предыдущих генов. Этот оператор формирования начальной популяции гарантирует допустимость сгенерированного решения относительно условий задачи.
3. Каждому гену, начиная с последнего случайным образом присваивается значение в диапазоне от 0 до , где – сумма всех последующих генов. Способ аналогичен предыдущему, с той только разницей, что заполнение начинается с конца.
4. Каждому гену случайным образом присваивается значение в диапазоне от 0 до [], где [] – целая часть числа. Данный оператор более или менее равномерно заполняет генотип, что может быть полезно для решения задач со входными данными определенного вида.

**Выбор родительской пары:**

ПАНМИКСИЯ Родительская пара выбирается случайным образом из всего поколения.

**Оператор скрещивания:**

В качестве оператора скрещивания выступает одноточечный кроссовер. Генотип родителей разрывается в одной и той же точке; потомок получается путем соединения части генотипа до точки разрыва первого родителя и части генотипа после точки разрыва второго родителя. Так, если – первый родитель, а , и – точка разрыва, то их потомком будет .

**Оператор мутации:**

Оператор мутации применяется к потомкам и заменяет значение гена, стоящего на случайной позиции, случайным другим значением из множества допустимых значений.

**Оператор селекции:**

В результате работы операторов поиска родительской пары и скрещивания генерируется множество потомков, количеством в два раза превосходящее множество родителей. Два этих множества объединяются в одно репродукционное множество и происходит отбор особей в новое поколение двумя операторами селекции (в зависимости от параметров запуска последовательно двумя, либо отдельно одним из них):

1. **Схема пропорциональной селекции. Особь репродукционного множества переходит в поколение в том случае, если приспособленность этой особи будет больше средней приспособленности множества . Поскольку размер каждого поколения фиксирован и одинаков, могут возникнуть случаи, когда в результате работы данного оператора следующее поколение формируется не полностью. В этом случае, доформирование производится либо случайным выбором из репродукционного множества, что дает возможность добора особей с малым значением функции приспособленности, в генотипах которых может содержаться часть генотипа оптимального решения; либо оператором вращения рулеточного колеса, что также сохраняет такую возможность.**
2. **Метод отбора путём вращения рулеточного колеса (стохастический выбор с возвращением). В первую очередь, вычисляется суммарная приспособленность особей, входящих в репродукционное множество : , где - мощность репродукционного множества, – приспособленность -й особи. Затем, для каждой особи находится вероятность по**падания в следующее поколение, пропорциональная приспособленности этой особи и обратно пропорциональная суммарной приспособленности: . Далее, вычисляется совокупная вероятность для каждой особи: . Случайным образом выбирается действительное число ∈ [0,], где . Если , то в популяцию копируется особь , иначе для копирования в популяцию выбирается особь , для которой выполняется условие . При использовании этого метода для формирования популяции выбор очередной копии производится каждый раз из всего множества решений, образующих репродукционное множество . При этом, одно и то же решение может быть репродуцировано несколько раз.

# Жадный алгоритм

Пусть есть предприятий. Для каждого предприятия есть возможных размеров вложений . Всего имеется средств, причем . Прибыль от вложения средств в -е предприятие обозначим .

Для сравнения с генетическим алгоритмом реализованы четыре варианта жадного алгоритма:

1. **Начиная с первого предприятия, на каждом шаге алгоритма определяется максимальная возможная прибыль от вложения оставшихся средств в** -е предприятие по формуле: , ∀, где .
2. **Аналогично второму варианту, однако просмотр начинается с конца.**
3. Начиная с первого предприятия, на каждом шаге находим наиболее выгодный размер вложений для предприятия. Для этого сравниваем удельные прибыли в диапазоне допустимого решения , ∀, где . Записываем индекс размера наиболее выгодного вложения в компоненту вектора решения, соответствующую данному предприятию:. На последнем шаге вкладываем все оставшиеся средства, т.е. .
4. **Аналогично третьему варианту, за исключением того, что просмотр и нахождение наиболее выгодного размера вложений начинается с последнего предприятия.**

В результате работы каждого из алгоритмов, получаем вектор решения, удовлетворяющий ограничению средств, представляющий какое-то решение.

## Эксперимент