Здравствуйте, меня зовут Пронина Елизавета Валерьевна. Тема дипломной работы:

*«Применение генетического алгоритма к задаче формирования инвестиционного портфеля»*

Научный руководитель: кандидат технических наук, доцент Неймарк Елена Александровна.

*Для начала познакомлю Вас с решаемой задачей, объектом исследования:*

**Задача формирования инвестиционного портфеля – это экономическая версия классической ЗОР.**

Классическая задача о ранце является задачей дискретной оптимизации. Впервые ЗОР была сформулирована Д. Данцигом и с тех пор находится в активном исследовании. Основные сферы применения находятся в областях управления экономическими, производственными и транспортными системами.

Предлагаемый метод решения задачи о ранце – **генетический алгоритм**. Данный алгоритм выбран в качестве **объекта исследования**.

**Актуальность** исследования предопределена широкой распространенностью и важностью различных прикладных проблем, решаемым генетическим алгоритмом.

**Цель моей работы заключается в следующем:**

* + Реализация генетического алгоритма для классической задачи о ранце.
  + Подбор параметров ГА для наиболее эффективного поиска решения задачи формирования инвестиционного портфеля.

*Как мы видим из названия работы*, для исследования ГА рассмотривается экономическая задача о ранце - **задача формирования инвестиционного портфеля**, где в роли цены акции выступает вес, а выгодностью акции является стоимость в классической задачи о ранце.

Оптимизация инвестиционного портфеля с помощью генетических алгоритмов повышает качество инвестирования финансовых средств в виде надеждого сбережения капитала или получения максимального дохода при допустимом риске.

*Вернемся к решаемой задаче.*

**Постановка задачи о ранце:**

*Представлена математическая постановка задачи о ранце.*

Пусть дано n предметов. Для каждого i-го предмета задан вес *> 0* и ценность  *> 0, i=1,.., n.* Задано ограничение на максимальный вес рюкзака ‒ . Каждый может принимать только одно из двух значений: *xi = 1*, если *i-*й предмет попадает в рюкзак, или*,* в противном случае. Требуется выбрать из заданного множества предметов набор с максимальной суммарной ценностью при одновременном соблюдении ограничения на суммарный вес найденного набора .

**Предлагаемый метод решения**:

*Как и говорилось в ведении, выбранным методом решения является – ГА:*

Генетические алгоритмы были предложены Джоном Генри Холландом в 1970 году и относятся к так называемым метаалгоритмам. Идея — составление алгоритмов поиска на основе биологической модели механизмов естественного отбора. Базовыми понятиями являются: популяция, отбор, мутация, скрещивание.

Плюсы ГА:

* Высокая скорость.
* Независимость от класса  исходной задачи (подразумевается, что скорость вычисления для люого вида задачи неизменна – линейная, единственная цель - получение максимальной функции приспособленности).

ГА представляют собой разновидность алгоритмов поиска и имеют преимущества перед другими алгоритмами при очень больших размерностях задач и отсутствия упорядоченности в исходных данных, когда альтернативой им является метод полного перебора вариантов.

***Главным достоинством ГА является то, что они могут применяться для решения сложных неформализованных задач, для которых не разработано специальных методов, т.е. ГА обеспечивают решение проблем.***

Минусы:

* Не гарантирует нахождение точного решения.

**Программная реализация ГА:**

На языке C# разработана программа, решающая классическую задачу о ранце генетическим алгоритмом. В ней реализованы следующие операторы:

1. 2 оператора начальной популяции (алгоритм Данцига, случайный алгоритм)
2. 3 оператора кроссовера (одноточечный, двуточечный и однородный кроссоверы)
3. 4 оператора мутации (точечная мутация, сальтация, инверсия, транслокация)
4. 1 оператор обработки ограничений (метод штрафных функций + линейное динамическое масштабирование)
5. 2 оператора селекции( линейная ранговая селекция, бетта-турнир) + модификация поколения

*Далее о классах тестовых задач, на которых проводились запуски генетического алгоритма:*

**Классы тестовых задач:**

Экономическая ЗОР имеет особенности в виде корреляций между выгодностью и ценой акций, поэтому исследование целесообразно проводить для 4 классов тестовых задач:

1. Задачи без корреляции (иллюстрируют примеры, когда вес и стоимость независимы).
2. Коррелированные задачи (одни из наиболее реалистичных в управлении,когда возврат инвестиций пропорционален вложенной сумме в некоторых небольших вариациях).
3. Сильнокоррелированные задачи (соответствуют реальной ситуации, где возвращается линейная функция от инвестиций).
4. Задачи с подсуммами (отражают ситуацию, когда прибыль каждого элемента является линейной функцией веса).

*Все классы тестовых задач программно реализованы.*

Задачи с подсумммами представляют собой такую ситуацию, когда есть несколько акций и над ними господствует другая акция, выйгрыш которой считается от выгрыша внутренних акций.

**Эксперимент**

Мною было замоделировано поведение эволюционного генетического алгоритма при помощи программной реализации. Для подбора параметров ГА, влияющих на наиболее эффективный поиск оптимального решения был реализован алгоритм проведения «исследования ГА».

* Количество особей.
* Количество итераций (поколений).
* Так как перебираются все возможные комбинации операторов генетического алгоритма, учитывается задание значения (для – турнира, по умолчанию = 2).
* Количество экспериментов для конкретной задачи из класса тестовых задач. Один эксперимент - запуск программы с различными комбинациями операторов генетического алгоритма (всего 48 комбинаций = 48 запусков ГА для одного эксперимента).
* Количество запускаемых задач для каждого класса.

Для одного запуска генетического алгоритма взяты следующие параметры:

* 30 особей
* 40 поколений
* = 14 (для бетта - турнирной селекции)
* Всего проведено 600 экспериментов (28800 запусков ГА), из них:
* 30 экспериментов на 1 задачу
* По 5 задач на конкретный класс тестовых задач
* Всего 4 класса тестовых задач

Для генерации задачи диапазон значений веса и цены - . Каждая задача прорешена программно **методом полного перебора**, что бы найти точное решение для исследования.

**Вывод результатов:**

Все отчеты представляют собой Excel таблицу. Для каждого эксперимента создается отдельный лист, в котором столбцы – комбинации операторов, а в строках выводится следующая информация:

* *Значение функции приспобленности для каждого поколения (всего 40).*
* *Максимальная функция приспособленности среди всех поколений.*
* *Количество итераций до получения лучшей особи.*

Последний лист – общий результат экспериментов по данной задаче. В нем содержится следующее:

* *Максимальное значение функций приспособленности среди всех экспериментов.*
* *Вероятность получения лучшей особи среди всех экспериментов****.(находится лучшее решение среди всех экспериментов, считаю количество тех эксперементов, у которых он встретился и делю на кол-во всех экспериментов).***
* *Среднее значение количества итераций до получения лучшей особи среди всех экспериментов.*

По всем сумммарным отчётам одного класса задач делается общий отчёт, в котором :

* *Минимальное отклонение значения функции приспособленности в процентном соотношении от оптимума (для нахождения оптимума реализован метод полного перебора).*
* *Средняя вероятность получения лучшей особи в процентах* ***(среди всех задач находится лучшее решение и считается кол-во задач, у которых это решение достигнуто и делится на количество всех задач).***
* *Cреднее количество итераций среди всех задач в процентом соотношении от наибольшего среднего количества итераций до получения лучшей особи (средняя скорость до наилучшего решения).* ***среди всех задач находится лучшее решение и складывается средн. кол – во итераций у задач, которых это решение достигнуто, делится на кол-во всех задач).***

**Анализ результатов:**

*Эффективные (лучшие)* комбинации – комбинации операторов, влияющих на наиболее эффективный поиск решения.

Из полученых результатов исследования для каждого класса тестовых задач выбрано по 5 эффективных комбинаций. Критериями отбора являлись:

* Вероятность нахождения глобального оптимума
* Скорость нахождения глобального оптимума

Для всех классов тестовых задач выбраны три эффективные комбинации. Критерий отбора - количество классов, содержащих комбинацию.

Данные результаты так же проанализированы и сделаны выводы, подтверждающие теоретический материал:

* Относительно ГА: Во всех оптимальных комбинациях в качестве оператора селекции выступает линейная ранговая схема. Это подтверждает теорию о том, что данная схема предотвращает преждевременную сходимость и приводит к наиболее лучшему решению.
* Относительно классов задач: Для классов тестовых задач с корреляцией и с сильной корреляцией вероятность нахождения глобального оптимума не превышает 22,67%, что подтверждает теорию о труднорешаемости таких задач, напротив, для классов тестовых задач без корряции и с подсуммами вероятность нахождения глобального оптимума достигает 72% и 84,67%, соответственно, что говорит о менее трудном нахождении их решения.

**Заключение:**

*В заключении хотелось бы сказать о* **практической значимости исследования:**

Она состоит в том, что подобраные эффективные наборы параметров генетического алгоритма для задачи формирования инвестиционного портфеля ускорят процесс и дадут наиболее точное решение для выбора самых выгодных акций по ограниченной цене.

Так как проблемы, решаемые генетическим алгоритмом являются актуальными и по сей день, в дальнейшем можно продолжить его исследование исходя из других свойств:

* Метод кодировки решений.
* Настройки параметров, такие как:
* количество особей,
* количество итераций.
* Частный критерий успеха(функция приспособленности).

Сложность функции приспособленности растет от сложности решаемой задачи

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

В [теории алгоритмов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%B2) **классом NP** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB.) *non-deterministic polynomial*) называют множество [проблем разрешимости](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BC%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8), решение которых возможно проверить на [машине Тьюринга](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_%D0%A2%D1%8C%D1%8E%D1%80%D0%B8%D0%BD%D0%B3%D0%B0) за время, не превосходящее[полинома](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BD%D0%BE%D0%BC) от размера входных данных, при наличии некоторых дополнительных сведений (так называемого *сертификата решения*).

Эквивалентно класс NP можно определить как содержащий задачи, которые можно за полиномиальное время решить на [недетерминированной машине Тьюринга](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B4%D0%B5%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0_%D0%A2%D1%8C%D1%8E%D1%80%D0%B8%D0%BD%D0%B3%D0%B0).

Задачи, имеющие полиномиальные по времени алгоритмы решения, можно решать с помощью компьютера значительно быстрее, чем путём прямого перебора, время которого [экспоненциально](https://ru.wikipedia.org/wiki/%D0%AD%D0%BA%D1%81%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B0). Это обуславливает практическое значение проблемы о [равенстве классов P и NP](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B2%D0%B5%D0%BD%D1%81%D1%82%D0%B2%D0%BE_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%B2_P_%D0%B8_NP).

**NP-полная задача** — в [теории алгоритмов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%BE%D0%B2) [задача с ответом «да» или «нет»](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B1%D0%BB%D0%B5%D0%BC%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B5%D1%88%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8) из [класса NP](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_NP), к которой можно свести любую другую задачу из этого класса за [полиномиальное время](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BD%D0%BE%D0%BC%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D0%B2%D1%80%D0%B5%D0%BC%D1%8F) (то есть при помощи операций, число которых не превышает некоторого полинома в зависимости от размера исходных данных). Таким образом, NP-полные задачи образуют в некотором смысле подмножество «типовых» задач в классе NP: если для какой-то из них найден «полиномиально быстрый» алгоритм решения, то и любая другая задача из класса NP может быть решена так же «быстро».

**Доказательство принадлежности к NPH**

Сведем [задачу о сумме подмножества](http://neerc.ifmo.ru/wiki/index.php?title=NP-%D0%BF%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B0_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8_%D0%BE_%D1%81%D1%83%D0%BC%D0%BC%D0%B5_%D0%BF%D0%BE%D0%B4%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B0) к задаче о рюкзаке. Пусть f\!\!:(S,s) \to (P,c,p) - функция, осуществляющее сведение. Она будет устроена так:

f(S,s) = ((S,S),s,s),

То есть, для каждого числа q \in S создадим предмет (q,q) с весом и стоимостью, равными значению числа q. А значения c и p возьмем равными s.

* Очевидно, f~ работает за полиномиальное от длины входа время.
* Если исходная [задача о сумме подмножества](http://neerc.ifmo.ru/wiki/index.php?title=NP-%D0%BF%D0%BE%D0%BB%D0%BD%D0%BE%D1%82%D0%B0_%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8_%D0%BE_%D1%81%D1%83%D0%BC%D0%BC%D0%B5_%D0%BF%D0%BE%D0%B4%D0%BC%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B0) имела решение S', то набор пар P' с весами, равными числам из S', будет решением задачи о рюкзаке.
* В обратную сторону - аналогично.