

Курсовая работа по теме:  
Генетические алгоритмы

Выполнили:  
студенты группы 3630102/70201  
Дамаскинский К.  
Колесник В.  
Пестряков Д.  
Рыженко В.

4 декабря 2019 г.

# Оглавление

<b>Введение</b>	<b>2</b>
<b>1 Генетические алгоритмы: понятийный аппарат, принцип работы</b>	<b>3</b>
1.1 Постановка задачи . . . . .	3
1.2 Описание алгоритма . . . . .	3
1.3 Операторы выбора родителей . . . . .	4
1.4 Операторы скрещивания . . . . .	5
1.5 Операторы мутации . . . . .	6
1.6 Операторы отбора . . . . .	6
1.7 Выбор параметров генетического алгоритма . . . . .	7
<b>2 Модернизация генетических алгоритмов</b>	<b>8</b>
<b>3 Преимущества и недостатки генетических алгоритмов</b>	<b>9</b>
<b>4 Примеры решения задач</b>	<b>10</b>
4.1 Задача об оптимальном управлении грузовым составом . . . . .	10
4.1.1 Описание проблемы . . . . .	10
4.1.2 Постановка задачи . . . . .	11
4.1.3 Формализация . . . . .	11
4.2 Задача круглого раскроя . . . . .	13
4.2.1 Описание задачи . . . . .	13
4.2.2 Математическая постановка задачи . . . . .	13
<b>Список литературы</b>	<b>14</b>

# Введение

Во все времена умы человека занимал вопрос о том, по каким законам развивается все сущее. В том числе, и живая природа. Работы Чарльза Дарвина, воспринимаемые ныне как данность, в свое время породили множество споров, не утихающих и по сей день. Ведь именно этот именитый британец выдвинул довольно смелое утверждение о том, что в основе эволюционного процесса лежат всего три вещи: наследственная изменчивость, борьба за существование, естественный отбор. Сколько бы ни разгоралось споров по поводу истинности работ Дарвина, именно эти три принципа красной нитью проходят через теорию построения *генетических алгоритмов*. Этот метод, как и остальные, не появился на пустом месте. Повышение вычислительной мощности в XX веке позволило ученым заняться моделированием эволюционных процессов. В 50-х годах были опубликованы циклы работ с результатами таких исследований. Еще через десятилетие инженеры решили применить данные принципы при решении некоторых прикладных задач. И уже в 70-х годах проводятся конференции, на которых обсуждаются проблемы и достижения в области изучения генетических алгоритмов как способа решения оптимизационных задач.

В данной работе мы рассмотрим общие идеи и подходы к решению задач оптимизации с использованием генетических алгоритмов. В ходе этого рассмотрения будут даны ответы на следующие вопросы:

1. Какими принципами руководствоваться при выборе размера популяции?
2. Каким образом сохранять размер популяции, если это возможно?
3. Как, исходя из условий задачи, выбрать критерий отбора?
4. Каким образом выбирается функция приспособляемости, она же функция цели?

Эту работу можно разбить на следующие части:

1. Генетические алгоритмы: понятийный аппарат, принцип работы. В этой части будут рассмотрены общие понятия и принципы построения генетических алгоритмов.
2. Модернизация генетических алгоритмов. В этой части будут рассмотрены основные часто встречающиеся проблемы, возникающие при решении задач с использованием генетических алгоритмов, и методы их решения.
3. Преимущества и недостатки генетических алгоритмов. В этой части будут рассмотрены границы применимости генетических алгоритмов. То есть будет дан ответ на вопрос: "Когда и почему стоит или не стоит применять генетические алгоритмы?"
4. Примеры решения задач. В этой части будут поставлены и формализованы две оптимизационные задачи, которые можно решить с применением генетических алгоритмов. Будут приведены их программные реализации.

# Глава 1

## Генетические алгоритмы: понятийный аппарат, принцип работы

### 1.1 Постановка задачи

Имеются:

- **Целевая функция** –  $f(x)$
- **Множество**, на котором задана функция –  $\Omega$

Ставится **оптимизационная задача** – найти такой элемент  $x_*$ , для которого выполняется одно из следующих неравенств:

$$f(x_*) \leq f(x), \forall x \in \Omega (\text{задача поиска минимума}),$$

$$f(x_*) \geq f(x), \forall x \in \Omega (\text{задача поиска максимума}).$$

В дальнейшем будем считать, что стоит задача **поиска минимума**.

### 1.2 Описание алгоритма

Генетические алгоритмы (далее ГА) относятся к стохастическим методам решения оптимизационных задач. ГА представляет собой адаптивный поисковый метод, основанный на селекции лучших элементов в популяции, подобно тому, как это происходит в теории Ч. Дарвина.

Введем основные понятия, применяемые в ГА:

- **Вектор** – упорядоченный набор чисел.
- **Хромосома** – вектор из каких-либо чисел. каждая компонента которого (*позиция хромосомы*) называется **геном**. Если вектор хромосомы представляет собой бинарную строку, то говорят о **бинарном генетическом алгоритме**. Если гены хромосомы являются вещественными числами, а их количество равно количеству переменных в целевой функции, то говорят о **генетическом алгоритме вещественного кодирования**. [2]
- **Особь** – вариант решения задачи, состоящий из 1 или нескольких хромосомы.
- **Скращивание** (**кроссинговер** – операция, при которой две хромосомы обмениваются своими частями.
- **Мутация** – случайное изменение одной или нескольких позиций в хромосоме.
- **Популяция** – совокупность особей.
- **Пригодность** (**приспособленность**) – функция, глобальный экстремум которой является решением задачи.
- **Эволюция популяции** – чередование поколений, в которых хромосомы изменяют свои значения так, чтобы каждое новое поколение наилучшим образом приспособлялось к внешней среде.

Предположим, что оптимизационная задача уже формализована и имеется способ представления решений в виде совокупности генов и хромосом. Для начала работы ГА генерируем множество случайных особей для начальной популяции. Далее приступаем к процессу размножения: попробуем на основе исходной популяции создать новую, так чтобы пробные решения в новой популяции были бы ближе к искомому экстремуму целевой функции. Критерием приспособленности особи является значение целевой функции: чем оно меньше, тем более приспособленной является особь (при поиске минимума). Следующим шагом в работе ГА являются мутации. Если скрещивание приводит к относительно небольшим изменениям пробных решений, то мутации могут привести к существенным изменениям значений пробных решений. После мутаций необходимо сформировать новую популяцию. Подобные действия повторяются итеративно, тем самым моделируется "эволюционный процесс". Через несколько поколений мы получим популяцию из похожих и наиболее приспособленных особей. Наилучшим образом адаптировавшуюся особь и будем считать решением нашей задачи.

Таким образом, генетический алгоритм работает по следующей схеме [3]:

1. Генерируем начальную популяцию из  $N$  особей, размер которой не будет меняться на протяжении работы всего алгоритма.
2. Выбираем пары особей-родителей оператором выбора родителя.
3. Проводим скрещивание каждой пары оператором скрещивания, производя 2 потомков.
4. Проводим мутацию потомков оператором мутации.
5. Формируем новую популяцию оператором отбора.
6. Повторяем шаги 2-5 пока не будет достигнут критерий окончания процесса.

Критерием окончания алгоритма может являться заданное количество поколений или *схождение* популяции. При схождении популяции все особи почти одинаковы и находятся в области некоторого экстремума. Скрещивание практически не изменяет популяцию, а вышедшие из области за счет мутации особи склонны вымирать, так как имеют меньшую приспособленность. Таким образом, схождение популяции обычно означает, что найден либо глобальный экстремум, либо локальный.

### 1.3 Операторы выбора родителей

Оператор выбора родителей считается эффективным, если он создает возможность перехода из одной подобласти поиска решения в другую. Это повышает вероятность нахождения глобального экстремума целевой функции. Существует несколько подходов к выбору родительских пар. Наиболее распространенными являются следующие:

#### \* *Панмиксия*

При данном методе выбора родителей каждому члену популяции случайным образом ставится в соответствие целое число на отрезке  $[1; N]$ , где  $N$  – количество особей в популяции. Данное число рассматривается как номер особи, которая примет участие в скрещивании. При таком выборе некоторые особи не будут участвовать в скрещивании, так как попадут в пару с самими собой. А некоторые особи могут участвовать в нескольких скрещиваниях одновременно.

#### \* *Инбридинг*

При данном методе первый родитель выбирается случайным образом, а вторым родителем является член популяции, ближайший к первому. *Ближайший* может пониматься, например, в смысле минимального евклидова расстояния между двумя вещественными векторами. Инбридинг можно охарактеризовать следующим свойством: так как для скрещивания выбираются только ближайшие особи, популяция может разбиться на группы вокруг подозрительных на экстремумы областей.

#### \* *Аутбридинг*

Данный метод отличается от инбридинга тем, что второй родитель выбирается максимально удаленным от первого. Аутбридинг направлен на сдерживание сходимости алгоритма к уже найденным решениям, заставляя алгоритм исследовать новые области.

#### \* *Селекция*

При данном методе родителями могут стать лишь те особи, значение приспособленности которых не меньше пороговой величины, например, среднего значения приспособленности по популяции.

Такой подход обеспечивает более быструю сходимость алгоритма. Однако для некоторых многомерных задач со сложным ландшафтом целевой функции быстрая сходимость может превратиться в преждевременную сходимость к квазиоптимальному решению. Этот недостаток может быть отчасти скомпенсирован использованием подходящего механизма отбора, который бы "тормозил" слишком быструю сходимость алгоритма (к примеру, 10% следующей популяции выбирать элитарным отбором, а остальные 90% создавать случайно).

\* **Турнирный отбор**

Турнирный отбор является вариацией селекции. Из популяции, содержащей  $N$  особей, выбирается случайным образом  $t$  особей, и лучшая из них записывается в промежуточный массив. Эта операция повторяется  $N$  раз. Особи в полученном массиве используются для скрещивания. Размер группы особей, отбираемых для одного раунда турнира, часто равен 2. Преимуществом данного способа является то, что он не требует дополнительных вычислений.

\* **Рулеточный отбор**

Рулеточный способ является вариацией селекции. Особи отбираются с помощью  $N$  "запусков" рулетки, где  $N$  - размер популяции. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер  $i$ -го сектора пропорционален вероятности попадания в новую популяцию  $P(i)$ , вычисляемой по формуле:

$$P(i) = \frac{f(i)}{\sum_{i=1}^N f(i)}$$

где  $f(i)$  - пригодность  $i$ -й особи. При таком отборе члены популяции с более высокой приспособленностью с большей вероятностью будут чаще выбираться, чем особи с низкой приспособленностью.

## 1.4 Операторы скрещивания

Оператор скрещивания применяют сразу же после оператора отбора родителей для получения новых особей-потомков.

\* **Дискретная рекомбинация**

Дискретная рекомбинация применяется к хромосомам с вещественными генами. Основными способами дискретной рекомбинации являются собственно дискретная рекомбинация и промежуточная рекомбинация.

– **Дискретная рекомбинация**

Дискретная рекомбинация соответствует обмену генами между особями. Создаются два потомка, каждый ген которых случайно с равной вероятностью выбирается среди двух соответствующих генов в той же позиции у особей-родителей.

– **Промежуточная рекомбинация**

В данном методе предварительно определяется числовой интервал значений генов потомков, который должен содержать значения генов родителей. Потомки создаются по следующему правилу:

$$\text{Потомок} = \text{Родитель 1} + \alpha * (\text{Родитель 2} - \text{Родитель 1}),$$

где множитель  $\alpha$  - случайное число на отрезке  $[-d; 1 + d]$ ,  $d \geq 0$ . Наиболее оптимальным считается значение  $d = 0.25$  [3]. Для каждого гена выбирается отдельный множитель.

\* **Кроссинговер**

Рекомбинацию бинарных строк принято называть кроссинговером.

– **Одноточечный кроссинговер**

Одноточечный кроссинговер моделируется следующим образом. Случайным образом определяется точка разрыва внутри хромосомы, в которой обе хромосомы делятся на две части и обмениваются ими. Такой тип кроссинговера называется одноточечным, так как при нем родительские хромосомы разделяются только в одной случайной точке.

- **Двухточечный кроссинговер**

При двухточечном кроссинговере выбираются 2 точки разреза. Родительские хромосомы обмениваются сегментами, ограниченными двумя точками. В настоящий момент считается, что двухточечный кроссинговер лучше, чем одноточечный. [3]

- **Многоточечный кроссинговер**

Для многоточечного кроссинговера выбирается случайно без повторений  $m$  точек разреза. Для получения двух потомков родительские особи обмениваются случайно выбранными сегментами, ограниченными точками разреза.

## 1.5 Операторы мутации

Оператор мутации необходим для "выбивания" популяции из локального экстремума и защиты от преждевременной сходимости. Это достигается за счет того, что изменяется случайно выбранный ген или несколько генов в хромосоме. Для мутации можно выбирать несколько особей из популяции, причем их число может быть случайным.

- \* **Плотность мутации**

Стратегия мутации с использованием понятия плотности заключается в мутировании каждого гена случайно выбранного потомка с заданной вероятностью. Величину вероятности применения мутации к каждому гену выбирают так, чтобы в среднем мутировало от 1 до 10% генов.

- \* **Двоичная мутация**

Для особей, кодированных двоичным кодом или кодом Грея, мутация заключается инвертировании случайно выбранного гена.

- \* **Другие виды мутации**

Для особи, представленной последовательностью генов, можно применить следующие операторы мутации: присоединение случайного гена, вставка случайного гена, удаление случайного гена, обмен местами случайно выбранных генов.

## 1.6 Операторы отбора

Оператор отбора необходим для создания новой популяции.

- \* **Отбор усечением**

При отборе усечением используют популяции особей-родителей и особей-потомков, отсортированные по возрастанию значения функции пригодности. Особи выбираются в соответствии с порогом  $T \in [0; 1]$ . Порог определяет, какая доля особей, начиная с самой пригодной, будет принимать участие в отборе. Среди особей, попавших "под порог" случайным образом выбирается одна особь и записывается в новую популяцию. Так повторяется  $N$  раз, пока размер новой популяции не станет равен старому. Новая популяция будет состоять из особей с высокой пригодностью, причем некоторые особи могут встречаться несколько раз, а самые пригодные могут и не попасть в популяцию.

- \* **Элитарный отбор**

При элитарном отборе используют популяции особей-родителей и особей-потомков. В новую популяцию выбираются  $N$  самых пригодных особей. Иногда данный метод комбинируют с другим – выбирают в новую популяцию 10% самых пригодных особей, а остальные 90% выбирают одним из методов селекции. Иногда эти 90% создают случайно, как на старте алгоритма. Преимуществом данного метода является то, что не допускается потеря лучших решений.

- \* **Отбор вытеснением**

В отборе вытеснением выбор особи в новую популяцию зависит от величины ее пригодности и от того, есть ли уже в формируемой популяции особь с аналогичным хромосомным набором. Из всех особей с одинаковой приспособленностью предпочтение отдается особям с разными генотипами, то есть тем особям, расстояние между которыми максимально. При данном методе не только сохраняются лучшие решения, но и поддерживается генетическое разнообразие. Отбор вытеснением наиболее пригоден для многоэкстремальных задач, при этом имеется возможность кроме глобального экстремума выделить локальные экстремумы, значения которых близки к глобальному.

## 1.7 Выбор параметров генетического алгоритма

Результат работы генетического алгоритма сильно зависит от того, каким образом настроены его параметры [1]. Основными параметрами ГА являются:

- длительность эволюции (количество поколений);
- размер популяции;
- оператор выбора родителей и его параметры;
- оператор скрещивания и его параметры;
- оператор мутации и его параметры;
- оператор отбора и его параметры.

Различные параметры ГА влияют на разные аспекты эволюционного поиска. Выделяют два наиболее общих:

1. Исследование пространства поиска.
2. Использование найденных "хороших" решений.

Основная цель в настройке параметров ГА и, одновременно, необходимое условие для стабильного получения хороших результатов работы алгоритма – это достижение баланса между исследованием пространства поиска и использованием найденных решений. Взаимосвязь между параметрами генетического алгоритма, а также их влияние на эволюционный процесс носит сложный характер, поэтому подбор параметров является нетривиален и определяется конкретной задачей.

Канонический ГА имеет следующие характеристики:

- целочисленное кодирование;
- все хромосомы имеют одинаковую длину;
- рулеточный отбор;
- одноточечный кроссинговер;
- двоичная мутация;
- новое поколение формируется только из особей-потомков.



## Глава 2

# Модернизация генетических алгоритмов

При использовании генетических алгоритмов для решения задач оптимизации могут возникать проблемы преждевременной сходимости. То есть можно попасть в локальный оптимум и не выйти из него, в силу того, что мы исчерпали возможности популяции к увеличению разнообразия потомства. Существует несколько модернизаций генетических алгоритмов, призванных избегать этих проблем. Рассмотрим некоторые из них:

1. При использовании популяции с малым числом особей гены распространяются слишком быстро, то есть особи становятся слишком похожими. Можно решить эту проблему тремя способами:
  - \* **Увеличение числа особей.** Это приведет к использованию дополнительной памяти, но метод весьма эффективен при использовании на простых функциях цели.
  - \* **Самоадаптация алгоритмов.** Это чаще всего используемый подход. Он позволяет использовать малый размер популяции. Идея основана на изменении значения вероятности мутации в зависимости от скрещивающихся особей, за счёт чего достигается самоуправляемость алгоритма. Такой подход называется *динамическими мутациями*.
  - \* **Создание массива для хранения особей, генотип которых мы утрачиваем во время формирования новых поколений.** Это, как и в первом случае, приведет к использованию дополнительной памяти, но позволит добавлять особей, которые могли быть не очень приспособленными ранее, но способных сейчас дать лучше адаптированное к новым условиям потомство. Или же позволит увеличить количество плохих генов, чтобы выйти из локального оптимума.
2. Перечисленные выше способы решения проблемы преждевременной сходимости приводят нас к другой – сохранению размеров популяции. Как видно, не во всех случаях представляется возможным держать константное для всех поколений число особей. Но, как можно заметить, применение подхода с самоадаптирующимися алгоритмами представляется наиболее приемлемым, если мы хотим решить сразу обе проблемы.

## Глава 3

# Преимущества и недостатки генетических алгоритмов

Для начала рассмотрим ряд *преимуществ* генетических алгоритмов:

1. Не требуют никакой информации о поведении функции (к примеру, дифференцируемости или непрерывности). Это дает возможность рассматривать ГА как метод решения задачи оптимизации, если для функции тяжело установить ее поведение.
2. Относительно стойки к попаданию в локальные оптимумы. Как было показано ранее, существуют модернизации ГА, позволяющие избежать преждевременной сходимости.
3. Могут быть использованы при решении обширного класса оптимизационных задач. ГА – это стохастический метод, основанный на эволюционном принципе. Поиск оптимума в множестве зачастую сродни поиску более приспособленных к каким-то условиям особей в природе.
4. Чаще всего просты в реализации. ГА – это стохастический метод с относительно простой структурой построения (популяция, скрещивание, отбор, функция цели), основанной на эволюционном принципе.
5. Можно использовать в задачах с изменяющейся средой. Это представляется возможным в силу того, что ГА по сути являются направленным перебором, результат которого напрямую не зависит от начальных условий, то есть под этим нужно понимать, что существует проблема преждевременной сходимости, которую с помощью некоторых модернизаций можно решить. Но при этом результат перебора зависит от функции цели. Поэтому ее можно менять на ходу.

Теперь перейдем к *недостаткам*:

1. Неразумно применять на долго вычисляемых функциях. Какими бы мощными не были вычислительные системы, время всегда будет дорогостоящим ресурсом. С учетом необходимости вычислять функцию в каждой популяции для каждой особи, ГА становится очень времязатратным.
2. Проблема представления данных в генетическом виде. Хотя чаще всего ГА просты в реализации, всё же существует ряд задач, для которых тяжело придумать генотип или оператор скрещивания. При этом упрощение модели может привести к потере части оптимальных решений.
3. Не для всех задач можно найти оптимальное кодирование генов. То есть не всегда существует простая модель, на которой можно было бы проводить быстрые вычисления.
4. Нет математически обоснованных критериев сходимости. Момент окончания работы алгоритма определяется исключительно из соображений здравого смысла.
5. Не всегда надежен. Если функция имеет множество локальных оптимумов или же она имеет изолированные точки, то ГА с большой долей вероятности может выдать неверное решение.

Как видно из анализа плюсов и минусов, ГА не являются универсальным методом решения всех проблем. К тому же, часть их преимуществ зачастую порождает множество проблем, которые нужно решать заранее при построении алгоритма. Но все они относятся к интеллектуальной трудоёмкости, нежели вычислительной, так как приходится учитывать специфику каждой задачи, порождающую свои узкие места. Причем если сложность построения модели – это уже сама по себе проблема, то простота построения модели так же может иметь неблагоприятные эффекты на решение задачи.

## Глава 4

# Примеры решения задач

### 4.1 Задача об оптимальном управлении грузовым составом

#### 4.1.1 Описание проблемы

Люди часто думают, что управлять поездом проще, чем машиной – руля же нет, всё едет само по себе. Только за сигналами следи да чаёк попивай. Однако на деле всё оказывается не так радужно.

При ведении грузового состава машинист сталкивается с целым рядом неординарных задач, требующих адекватной оценки ситуации, быстрой реакции, аналитического склада ума и, зачастую, хорошей интуиции. Давайте обратим внимание на факторы, влияющие на процесс движения грузового состава.

##### 1. Длина поезда.

Длина грузового состава может достигать нескольких километров. Из-за этого при изменении скорости движения – торможении и разгоне – на автосцепки вагонов, находящихся в начале, середине и конце действуют разные силы.

Представим ситуацию: поезд шёл под уклон, затормозил, а дальше начался затяжной подъём. Машинист собирает схему на тягу, локомотив начинает тянуть за собой поезд.

Первые вагоны уже не удерживаются тормозом, чего нельзя сказать про задние. Таким образом, мы имеем неиллюзорный шанс порвать автосцепку в конце поезда. Локомотивная бригада скорее всего не увидит оторвавшийся хвост состава, что чревато самыми неприятными последствиями.

##### 2. Распределение массы вдоль поезда.

Здесь проблема носит тот же характер, что и в предыдущем пункте: если оставить лёгкие вагоны впереди, а тяжёлые сзади, то при резком старте, скорее всего, произойдёт разрыв там, где кончаются пустые и начинаются гружёные вагоны. Очень нехорошая ситуация может сложиться при входе на подъём с равнинного участка: пусть, скажем, первая половина поезда порожняя, а вторая гружёная. Тогда машинист может, легко втащив на подъём первую половину вполсилы, добавить позиций, чтобы затянуть вторую. В такой ситуации та же самая сцепка – на стыке пустых и гружёных вагонов – получит просто фантастическую нагрузку.

##### 3. Погодные условия.

Здесь ситуация схожа с той, которую мы наблюдаем при попытке стронуться с места на завязшем в трясине автомобиле – момент, подводимый к колесу от двигателя, оказывается больше момента, с которым сила трения покоя действует на колесо, и в результате начинается буксование. На железной дороге буксование можно встретить в куда более простых условиях – достаточно сильного дождя и слишком тяжёлого поезда.

Но если параметры локомотива на этапе сборки состава подбираются таковыми, чтобы он гарантированно мог стронуть с места поезд в любых погодных условиях, то о торможении уже приходится думать машинисту – если слишком резко "дать по тормозам", то начнётся буксование и воздух в магистрали очень быстро закончится. Состав станет неуправляемым.

Описав основные проблемные ситуации, мы обнаружили наиболее уязвимые узлы управления: автосцепка и тормоз.<sup>1</sup>

---

<sup>1</sup> **Кратко о работе поездного тормоза** Принцип работы следующий: воздух закачивается компрессором в тормозные резервуары под большим давлением. При необходимости затормозить воздух с задаваемой машинистом интенсивностью вытравливается из резервуара в общую тормозную магистраль, ответвления от которой подведены непосредственно

### 4.1.2 Постановка задачи

На вход даются следующие параметры:

- Максимальная сила тяги, которую способен развить локомотив
- Предельная нагрузка на автосцепку
- Погодные условия
- Предельный коэффициент трения покоя при данных погодных условиях
- Зависимость силы торможения от давления в магистрали
- Скорость сброса (набора) воздуха в магистрали в разных положениях тормозного крана <sup>2</sup>
- Длина поезда
- Время прохождения "тормозной волны" вдоль одного вагона (время, в течение которого давление ТМ в данном вагоне сравнивается с давлением в ТМ соседнего вагона)
- Распределение массы поезда
- Профиль пути

Требуется предоставить режим движения, при котором будут выполнены следующие условия:

- Поезд доедет до пункта назначения в целости за наименьшее время
- На каждый следующий участок пути поезд подходит с максимальным давлением в ТМ и скоростью не выше максимально допустимой

Запас топлива считаем неограниченным – обычно в реальных условиях с этим действительно нет проблем.

Погода в течение всего маршрута следования считается неизменной (ясно, что если погодные условия изменились, можно разбить путь на части, на которых погодные условия постоянны).

Будем считать, что в каждый момент времени поезд целиком находится на участке пути с одним профилем – с точки зрения анализа опасности ситуации такое приближение недопустимо (мы не можем понять, что будет в ситуации разрыва поезда), но с точки зрения численных расчётов мы не внесём ложных смягчений режима вождения.

### 4.1.3 Формализация

Вход

- $F_{\text{тяги max}}$  – предельная сила тяги локомотива,  $\kappa H$
- $F_{\text{СА max}}$  – предельная нагрузка на автосцепку,  $\kappa H$
- $W$  – условная единица, характеризующая погодные условия, численно обозначающая степень увлажнения рельса
- $\mu_{\text{max}}(W)$  – зависимость предельного коэффициента трения покоя колеса о рельс от погодных условий
- $F_{br}(P_{\text{ТМ}})$  – зависимость силы торможения от давления в тормозной магистрали (далее ТМ),  
 $[F_{br}] = \kappa H, [P_{\text{ТМ}}] = \kappa \text{Па}$
- $P_{\text{ТМ max}}, P_{\text{ТМ min}}$  – максимальное и минимальное допустимые давления в ТМ
- $V$  – число вагонов в поезде
- $l_v$  – длина вагона,  $m$ . Полагаем, что все вагоны имеют одинаковую длину
- $\tau$  – время распространения тормозной волны вдоль одного вагона,  $s$

к тормозным колодкам. Соответственно чем выше давление в магистрали, тем сильнее прижимаются колодки к колесу и тем быстрее происходит торможение. При отпуске тормоза воздух из магистрали выпускается в атмосферу. Одновременно включается компрессор и воздух нагнетается в тормозные резервуары заново. В данной задаче важно, что это достаточно длительная процедура

<sup>2</sup>Под **силой торможения** будем понимать силу, с которой тормозная колодка прилегает к колесу.

- $v_{TM}(S) = \frac{dP}{dt}(S)$  – зависимость скорость стравливания (или набора) воздуха из тормозной магистрали от выбранной машинистом позиции крана  $S$ ,  $\kappa Pa/c$ ,  $S = \overline{1,6}$
- $m(n)$  – распределение массы поезда от номера вагона, *тонн*,  $n = \overline{1, V}$
- $\{(v_{max}^{(k)}, \alpha^{(k)}, d^{(k)})^T\}_{k=\overline{1, M}}$  – профиль пути. Задаётся в виде трёхкомпонентных векторов, состоящих из предельной скорости на участке (*км/ч*), угла наклона (*радианы*) и длины (*м*).

## Выход

$(\{U^{(k)}\}^h, \{S^{(k)}\}^h, h, N)^T_{k \in \mathbf{N}}$  – кортеж, состоящий из табличных функций (1, 2 компоненты), заданных на  $\{t_j\}^h, j = \overline{1, N} : t_i = hi, t_N = T_0$ . Первая компонента соответствует доле от максимальной тяги, а вторая – позиции тормозного крана. Третья и четвёртая компоненты кортежа – параметры временной сетки, на которой определены функции. [4]

## Ограничения

На  $k$ -ом участке перегона:

$$\int_0^{T_0} v_{TM}(t) dt = 0 \quad (1)$$

$$|F_i(t) - F_{i-1}(t)| \leq F_{CAmax}, \forall i = \overline{2, V} \forall t \in \{t^h\} \quad (2)$$

$$v_k + \frac{1}{m(1)} \int_0^{T_0} F_1(t) dt \leq v_{k+1} \quad (3)$$

$$F_{тяги}(t) \leq F_{тяги \max}, \forall t \in \{t^h\} \text{ (временно выкидываем, тяга конст)} \quad (4)$$

$$\int_0^{T_0} v(t) dt = d^{(k)} \quad (5)$$

$$F_{тяги}(t) \cdot v_{TM}(t) \leq 0 \forall t \in \{t^h\} \quad (6)$$

(1) – давление в ТМ не должно измениться к концу перегона

(2) – в каждый момент времени нагрузка на автосцепку не должна превышать предельную

(3) – скорость на выходе не должна превышать ограничение на следующем участке. Так как все вагоны движутся с одиноковой скоростью, можно не умаляя общности рассматривать первый вагон

(4) – тепловоз не может развить мощность, бóльшую, чем конструкционная

(5) – за данный промежуток времени тепловоз должен пройти заданное расстояние

(6) – в каждый момент либо происходит набор воздуха в ТМ вместе с набором тяги, либо торможение с выключенной тягой

Кроме того, для упрощения задачи будем блокировать ручку тормозного крана до тех пор, пока тормозная волна не дойдёт до конца поезда.

## Функция цели

$$T_0 \rightarrow \min$$

## Алгоритм решения

Мы разобьём общую задачу – нахождение оптимального режима на всём пути – на подзадачи нахождения оптимального режима на каждом отдельном участке (будем пренебрегать возможными оптимизационными манёврами на стыках профиля ввиду сложности).

Данную задачу концептуально мы будем решать следующим образом. В генетический алгоритм будет передаваться желаемое время прохождения перегона  $T_0$ . Алгоритм будет пытаться найти режим движения, в котором поезд сможет благополучно дойти до пункта назначения за данное время. Параметр  $T_0$ , исходя из результатов работы генетического алгоритма, будет корректироваться по методу половинного деления. Начальный  $T_0$  мы найдём аналитически – вычислим время, за которое поезд гарантированно сможет преодолеть перегон с указанными выше условиями.

Таким образом, наша задача будет решаться связкой генетического алгоритма и метода половинного деления. За счёт такого подхода мы сможем использовать все достоинства ГА и при этом обойти его недостатки: с помощью ГА мы будем отвечать на вопрос, **можно ли** подобрать режим ведения так, чтобы за данное  $T_0$  поезд успешно прошёл участок, а не **какое  $T_0$  оптимальное**. Сам параметр  $T_0$  же будет вычисляться методом половинного деления, сходимость которого нам точно известна.

Мы полагаем, что такой подход лучше, поскольку задавая ГА директивный вопрос: *да* или *нет*, мы имеем более обоснованную надежду на то, что ГА сможет дать корректный ответ, чем задавая вопрос: *какое значение оптимально*, хотя бы потому, что область возможных решений несравнимо меньше.

## 4.2 Задача круглого раскроя

### 4.2.1 Описание задачи

Рассматриваемая задача — поиск рационального плана раскроя плоского листа на предметы круглой формы. Задачи такого рода впервые были поставлены еще в 1940-х академиком Л.В. Канторовичем. С тех пор появилось большое количество новых постановок и методов решения. Однако существуют практически значимые постановки задач и технологические ограничения, для которых решение задачи раскроя и разработка новых методов решения по-прежнему актуальны. Вообще, задача плоского раскроя — это оптимизационная задача поиска наиболее плотного размещения множества меньших по размеру плоских предметов, деталей, на больших объектах, заготовках

### 4.2.2 Математическая постановка задачи

Заданы следующие параметры и условия:

- Полубесконечная полоса ширины  $W$
- $n$  круглых деталей
- Радиусы деталей  $r_i, i = \overline{1, n}$
- Детали попарно не пересекаются:  

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i - r_j)^2, i, j = \overline{1, n}, i \neq j$$
- Детали не выходят за границы полосы: 
$$\begin{cases} x_i - r_i \geq 0 \\ y_i - r_i \leq 0, \\ y_i + r_i \leq W \end{cases} \quad i = \overline{1, n}$$

Здесь  $(x_i, y_i), i = \overline{1, n}$  - координаты центров деталей

Необходимо разместить детали на полосе так, чтобы занимаемая часть полосы была минимальна, т. е.

$$\max_{i=\overline{1, n}} (x_i + r_i) \xrightarrow{(x_i, y_i)} \min$$

# Список литературы

- [1] Ю. Р. Цой В. Г. Спицын. *Представление знаний в информационных системах: учеб. пособие*. Томск: Изд-во ТПУ, 2007.
- [2] С. П. Дударов. *Математические основы генетических алгоритмов: учеб. пособие*. М.: РХТУ им. Д. И. Менделеева, 2012.
- [3] Т. В. Панченко. *Генетические алгоритмы*. Астрахань: Издательский дом «Астраханский университет», 2007.
- [4] *СА-3*. URL: <https://ru.wikipedia.org/wiki/%D0%A1%D0%90-3>.