

# 1 Элитарная муравьиная система

## 1.1 Отличия элитарной модификации от общего случая

Одной из ключевых проблем, с которой сталкивается муравьиный алгоритм, является положительная обратная связь. В условиях положительной обратной связи все муравьи могут сосредоточиться на одном неоптимальном пути, что приводит к застою и снижению качества решений. Чтобы избежать этой проблемы, была разработана элитарная модификация муравьиного алгоритма.

Элитарная модификация направлена на улучшение качества решений и ускорение сходимости алгоритма. Основная идея заключается в том, чтобы сохранить лучшие найденные решения (или пути) и обеспечить их более высокую вероятность выбора в последующих итерациях. Это позволяет алгоритму использовать уже найденные хорошие решения, что значительно повышает вероятность нахождения глобального оптимума.[2]

Причины применения элитарной модификации разнообразны. Во-первых, она позволяет улучшить качество решений, так как лучшие пути сохраняются и становятся более предпочтительными. Во-вторых, это ускоряет сходимость алгоритма, так как муравьи могут опираться на уже найденные эффективные маршруты. В-третьих, элитарная модификация помогает избежать ситуации, когда алгоритм застревает в локальных минимумах, так как лучшие решения продолжают оказывать влияние на выбор путей.[2]

Элитарная модификация отличается от общего случая муравьиного алгоритма несколькими ключевыми аспектами. Во-первых, в общем случае алгоритм не сохраняет лучшие решения, а просто обновляет феромоны на основе всех найденных путей. В элитарной модификации лучшие решения сохраняются и получают дополнительное внимание. Во-вторых, вероятность выбора пути в общем случае зависит от феромонов и эвристической информации, тогда как в элитарной модификации вероятность выбора лучших путей увеличивается. Наконец, обновление феромонов в элитарной модификации может происходить более агрессивно для лучших путей, что позволяет им сохранять свое влияние на выбор путей.

## 1.2 Формализация элитарной модификации

Правило обновления феромонов принимает вид:

$$\tau_{i,j}(t+1) = (1-p)\tau_{i,j}(t) + p\Delta\tau_{i,j,e}(t)$$

где  $(i, j)$  — край наилучшего маршрута (либо на текущей итерации, либо с начала алгоритма)

$$\Delta\tau_{i,j,e}(t) = \begin{cases} \frac{Q}{L^+}, & (i, j) \in T^+ \\ 0, & (i, j) \notin T^+ \end{cases}$$

”Элитные” муравьи выделяют феромоны только по рёбрам лучшего найденного маршрута  $T^+$ . Для поставленной задачи берётся значение феромона ”элитного” муравья на каждом ребре маршрута  $T^+$  равным  $\frac{Q}{L^+}$ , где  $L^+$  — длина маршрута  $T^+$ .

Шаг модифицируется следующим образом:  $k$ -й муравей перемещается с вероятностью  $q_0$  из города  $i$  в наиболее оптимальный город  $z \in J_{i,k}$  и с вероятностью  $(1-q_0)$  выбирает город  $j$  по шагу  $P_{i,j,k}(t)$ . Чем больше  $q_0$ , тем выше эффективность использования опыта, полученного колонией муравьёв при синтезе новых маршрутов. Наиболее привлекательный город определяется так:

$$z = \arg \max_{j \in J_{i,k}} (\tau_{i,j}(t))^\alpha (\eta_{i,j})^\beta$$

На каждой итерации при переходе из города  $i$  в город  $j$  муравей ”съедает” какое-то количество феромонов с этого ребра графа. Это ребро теряет свою привлекательность для других муравьёв: они начинают рассматривать другие маршруты. Решения становятся более разнообразными благодаря динамичному обновлению распределения феромонов.

Также:

1. Уровень феромонов на рёбрах обновляется не только на каждой итерации, но и при переходе муравьёв из узла в узел.
2. В конце итерации уровень феромонов повышается только на лучшем из найденных путей.
3. Муравей либо с определённой вероятностью выбирает лучшее ребро, либо производит выбор как в классическом алгоритме.

## 2 Мах-Мин муравьиная система

### 2.1 Отличия Мах-Мин модификации от общего случая

Одной из основных проблем, с которыми сталкивается классический муравьиный алгоритм, является неэффективное использование феромонов, что может привести к преждевременному застреванию в локальных минимумах. В ответ на эту проблему была разработана Мах-Мин модификация муравьиного алгоритма, которая направлена на улучшение качества решений и повышение устойчивости алгоритма к застоям.

Мах-Мин модификация основывается на принципе ограничения феромонов, что позволяет контролировать их уровень и избегать чрезмерного накопления. В этой модификации устанавливаются максимальные и минимальные границы для значений феромонов, что обеспечивает более равномерное распределение феромонов по всем путям. Это позволяет избежать ситуации, когда один путь становится доминирующим, и способствует более разнообразному поиску решений.[1]

Причины применения Мах-Мин модификации разнообразны. Во-первых, она позволяет улучшить качество решений, так как феромоны обновляются более сбалансированно, что способствует более равномерному исследованию пространства решений. Во-вторых, такая модификация помогает ускорить сходимость алгоритма, так как муравьи могут более эффективно использовать информацию о феромонах, не застревая на одном и том же пути. В-третьих, Мах-Мин модификация снижает вероятность преждевременного застревания в локальных минимумах, так как феромоны на менее популярных путях также имеют шанс быть выбраны.[1]

Отличия Мах-Мин модификации от общего случая муравьиного алгоритма заключаются в подходе к обновлению феромонов. В общем случае феромоны обновляются на основе всех найденных путей, что может привести к доминированию одного или нескольких путей. В Мах-Мин модификации феромоны ограничиваются, что позволяет избежать чрезмерного влияния отдельных путей на выбор муравьёв. Кроме того, в Мах-Мин модификации феромоны обновляются с учетом как максимальных, так и минимальных значений, что обеспечивает более сбалансированное распределение феромонов.

### 2.2 Формализация Мах-Мин модификации

Обновление феромонов в Мах-Мин задается следующим образом:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}^{best}$$

где

$$\Delta\tau_{ij}^{best} = \frac{1}{f(s_{best})}$$

где  $f(s_{best})$  обозначает длину решения, полученного либо от итерационного лучшего решения  $s_{ib}$ , либо от глобального лучшего решения  $s_{gb}$ .

В Мах-Мин используется только одно решение для обновления феромонов, что является важным средством для эксплуатации поиска. Выбор между итерационным лучшим решением  $s_{ib}$  и глобальным лучшим решением  $s_{gb}$  влияет на то, как история поиска будет использоваться. Использование только  $s_{gb}$  может привести к слишком быстрому сосредоточению поиска вокруг этого решения, что ограничивает исследование других возможных решений. В то время как использование  $s_{ib}$  позволяет избежать этого, так как итерационные лучшие решения могут значительно различаться от итерации к итерации.

Мах-Мин вводит явные ограничения на минимальные и максимальные значения феромонов:

$$\tau_{min} \leq \tau_{ij}(t) \leq \tau_{max}$$

После каждой итерации необходимо убедиться, что феромоны соответствуют этим ограничениям:

$$\text{если } \tau_{ij}(t) > \tau_{max}, \text{ то } \tau_{ij}(t) = \tau_{max}$$

$$\text{если } \tau_{ij}(t) < \tau_{min}, \text{ то } \tau_{ij}(t) = \tau_{min}$$

### 3 Заключение и выводы

- **nodes:**

- Список координат узлов (городов), представленный в виде кортежей.
- Пример:

$$\text{nodes} = [(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)]$$

где  $n$  — количество узлов.

- **colony\_size:**

- Размер колонии муравьев (количество муравьев, участвующих в поиске решения).
- Пример:

$$\text{colony\_size} = 5$$

- **elitist\_weight:**

- Вес, с которым обновляются феромоны на лучшем найденном маршруте в элитарном алгоритме.
- Пример:

$$\text{elitist\_weight} = 1.0$$

- **min\_scaling\_factor:**

- Минимальный коэффициент масштабирования для феромонов в Max-Min алгоритме.
- Пример:

$$\text{min\_scaling\_factor} = 0.001$$

- **alpha:**

- Коэффициент, контролирующий влияние феромонов на выбор следующего узла.
- Пример:

$$\alpha = 1.0$$

- **beta:**

- Коэффициент, контролирующий влияние эвристической информации на выбор следующего узла.
- Пример:

$$\beta = 3.0$$

- **rho:**

- Коэффициент испарения феромонов.
- Пример:

$$\rho = 0.1$$

- **phi:**

- Коэффициент, используемый для локального обновления феромонов.
- Пример:

$$\phi = 0.1$$

- **pheromone\_deposit\_weight:**

- Вес, с которым феромоны добавляются на рёбра, пройденные муравьями.
- Пример:

$$\text{pheromone\_deposit\_weight} = 1.0$$

- **initial\_pheromone:**

- Начальное значение феромонов на рёбрах.

– Пример:

`initial_pheromone = 1.0`

- **steps:**

– Количество итераций (шагов), которые будет выполнять алгоритм для поиска решения.

– Пример:

`steps = 100`

В основном блоке кода, который запускает алгоритм, параметры задаются следующим образом:

```
_colony_size = 5
_steps = 200
random.seed(10)
_nodes = [(random.uniform(0, 200), random.uniform(0, 200)) for _ in range(0, 50)]
```

- EAS может быстро находить хорошие решения, особенно на простых экземплярах TSP, благодаря использованию информации о всех муравьях. Однако, на более сложных экземплярах EAS может не всегда находить глобальный оптимум из-за риска застревания в локальных минимумах.[4]
- EAS может демонстрировать высокую эффективность на малых и средних задачах, но ее производительность может ухудшаться на больших и сложных экземплярах.[4]
- MMAS, благодаря своей структуре обновления феромонов, часто показывает более высокую эффективность в нахождении глобального оптимума, особенно на сложных экземплярах TSP. Ограничение феромонов помогает избежать чрезмерного накопления феромонов на менее оптимальных путях, что способствует более разнообразному поиску.
- MMAS может требовать больше итераций для достижения хороших решений, но в конечном итоге часто находит более качественные решения.

Из полученных результатов очевидно, что при данной постановке с большим количеством итераций для задачи средней сложности Max-Min модификация справляется лучше: этот алгоритм находит более короткий маршрут за отведенное время, и, очевидно, лучше сходится к оптимуму.

На более простой задаче элитарная модификация показывает лучший результат, причём с существенным отрывом, что подтверждает наши теоретические данные.

В ходе данной курсовой работы был проведен сравнительный анализ двух модификаций муравьиного алгоритма — элитарной модификации (EAS) и Max-Min модификации (MMAS) — в контексте решения задачи коммивояжера (TSP). Оба подхода продемонстрировали свою эффективность в поиске оптимальных решений, однако их характеристики и поведение в процессе поиска существенно различаются.

Элитарная модификация муравьиного алгоритма, благодаря использованию информации о всех найденных решениях, позволяет быстро находить хорошие маршруты, особенно на простых и средних экземплярах задачи. Однако, ее склонность к застреванию в локальных минимумах может ограничивать эффективность на более сложных задачах. Это делает EAS подходящим выбором для ситуаций, где требуется быстрое получение приемлемых решений, но не всегда гарантирует нахождение глобального оптимума.

С другой стороны, Max-Min модификация муравьиного алгоритма, с ее строгими ограничениями на феромоны и обновлением только на основе лучших найденных решений, демонстрирует более стабильную и надежную сходимость к глобальному оптимуму. Хотя MMAS может требовать больше итераций для достижения хороших результатов, его способность избегать локальных минимумов делает его более предпочтительным для сложных задач, где качество решения имеет первостепенное значение.

В результате проведенного анализа можно сделать вывод, что выбор между EAS и MMAS должен основываться на конкретных требованиях задачи. Для задач, где важна скорость нахождения решения, EAS может быть более подходящим вариантом. В то время как для задач, требующих высокой точности и надежности, MMAS представляется более эффективным выбором.