

Метод построения конечных автоматов на основе муравьиного алгоритма

Чивилихин Д.С., Ульяновцев В.И.
Научный руководитель – к.т.н. Царев Ф.Н.

II Всероссийский конгресс молодых ученых. НИУ ИТМО

11.04.2013

Конечный автомат

- КА – шестерка $\langle S, s_0, E, A, \delta, \lambda \rangle$;
- S – множество состояний;
- s_0 – начальное состояние;
- E – множество входных событий;
- A – множество выходных воздействий;
- $\delta: S \times E \rightarrow S$ – функция переходов;
- $\lambda: S \times E \rightarrow A^*$ – функция действий.

Задача построения автоматов

- X – множество КА с заданными параметрами:
 - ♦ N – число состояний;
 - ♦ E – множество входных событий;
 - ♦ A – множество выходных воздействий.
- $f: X \rightarrow R$ – **функция приспособленности (ФП)**
- b – барьерное значение f
- **Задача:** найти автомат x из X такой, что $f(x) \geq b$


Состояние проблемы

- Эволюционные стратегии
- Генетические алгоритмы
- Метод имитации отжига

Предлагаемый метод построения автоматов

1. Свести задачу построения автомата к оптимизации на графе
2. Использовать **муравьиный алгоритм** нового типа для решения задачи

Существующие муравьиные алгоритмы не подходят для решения задачи



Решение (1): представление пространства поиска в виде графа

- Граф:
 - ♦ вершины – конечные автоматы;
 - ♦ ребра – *мутации* конечных автоматов.
- Мутация – небольшое изменение структуры автомата:
 - ♦ изменение состояния, в которое ведет переход;
 - ♦ изменение действия на переходе.

Решение (2): муравьиный алгоритм

1. Граф = {случайный автомат}
2. While (true)

Построение решений муравьями

Обновление значений феромона

Проверка условий останова

Построение решений муравьями

- N муравьев
- Запускаются:
 - Последовательно: 1, 1, 1, 1, 2, 2, 2, 2
 - или «параллельно»: 1, 2, 1, 2, 1, 2, 1, 2

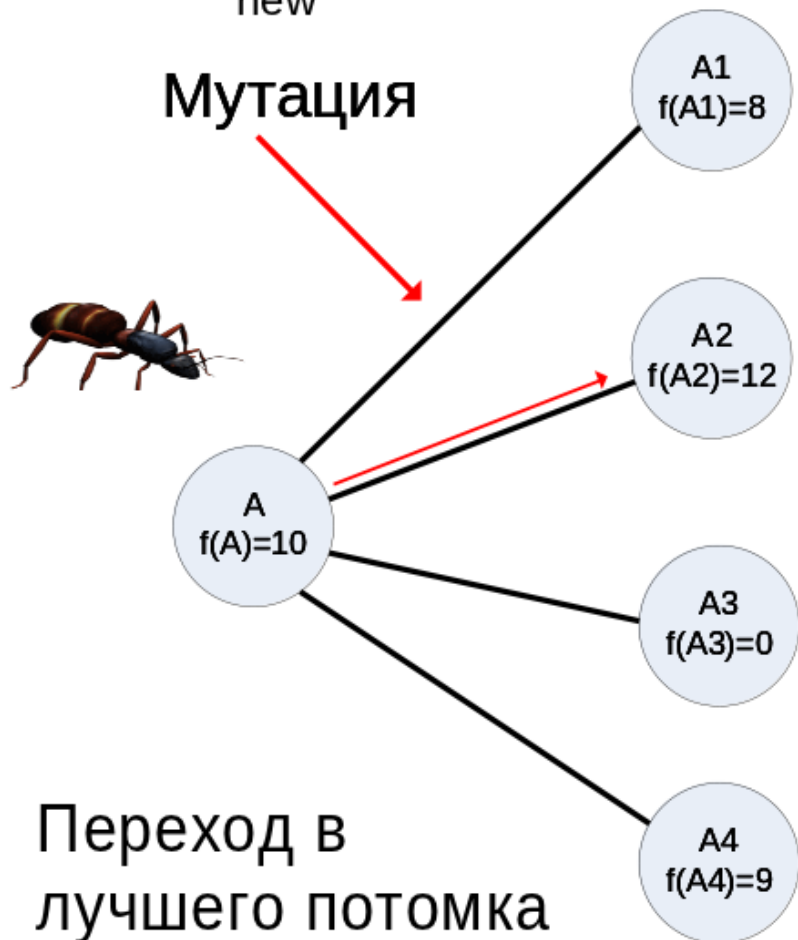
Алгоритм работы муравья

- Муравей помещается в вершину графа
- У каждого муравья – ограниченное число шагов
- Шаг муравья – переход в следующую вершину

Выбор следующей вершины

$$P = P_{\text{new}}$$

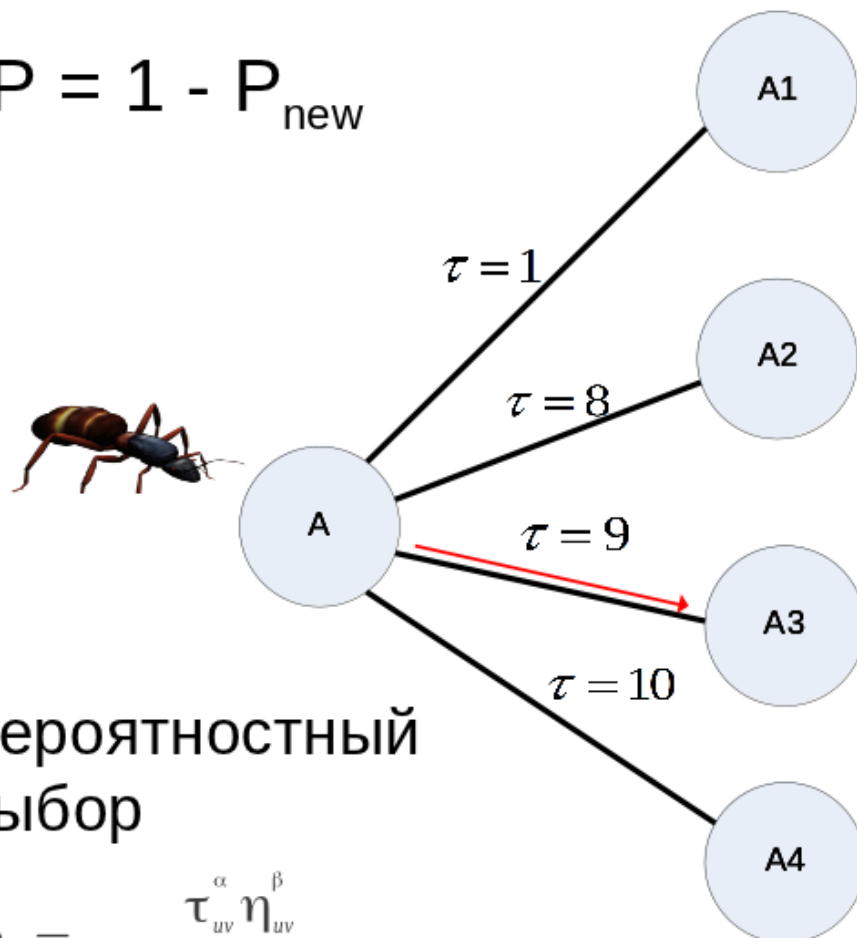
Мутация



$$P = 1 - P_{\text{new}}$$

Вероятностный выбор

$$p_{Av} = \frac{\tau_{uv}^{\alpha} \eta_{uv}^{\beta}}{\sum_{w \in \{A1, A2, A3, A4\}} \tau_{uw}^{\alpha} \eta_{uw}^{\beta}}$$



Обновление значений феромона

- Качество решения (пути муравья) – максимальное значение ФП вершины пути
- Обновление τ_{uv}^{best} – наибольшего значения феромона, отложенного на ребре (u, v)
- Новое значение вычисляется по формуле:

$$\tau_{uv} = \rho \tau_{uv} + \tau_{uv}^{best}$$

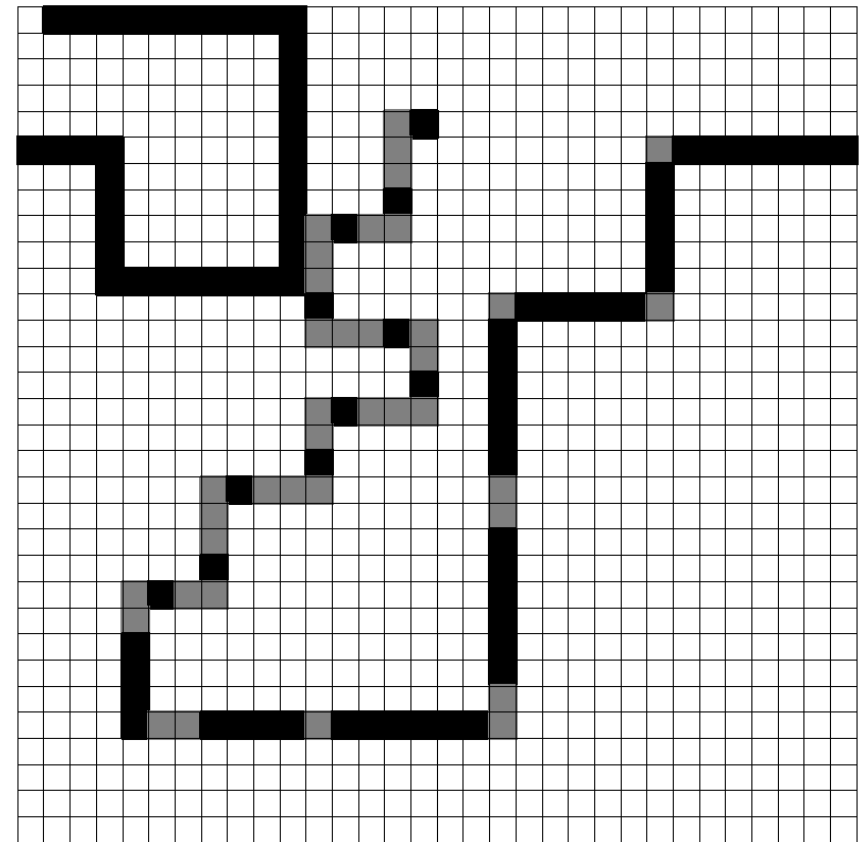
- $\rho \in [0, 1]$ – скорость испарения феромона

Отличие от классических муравьиных алгоритмов

- Классика:
 - Вершины — компоненты решений
 - Полные решения строятся муравьями
- Предложенный муравьиный алгоритм:
 - Вершины — полные решения
 - Муравьи перебирают полные решения

Пример применения: задача «Умный муравей»

- Поле – тор $N \times N$
- M клеток с едой
- K ходов
- Положение еды и начальная позиция муравья фиксированы
- Цель – создать муравья, который съест всю еду

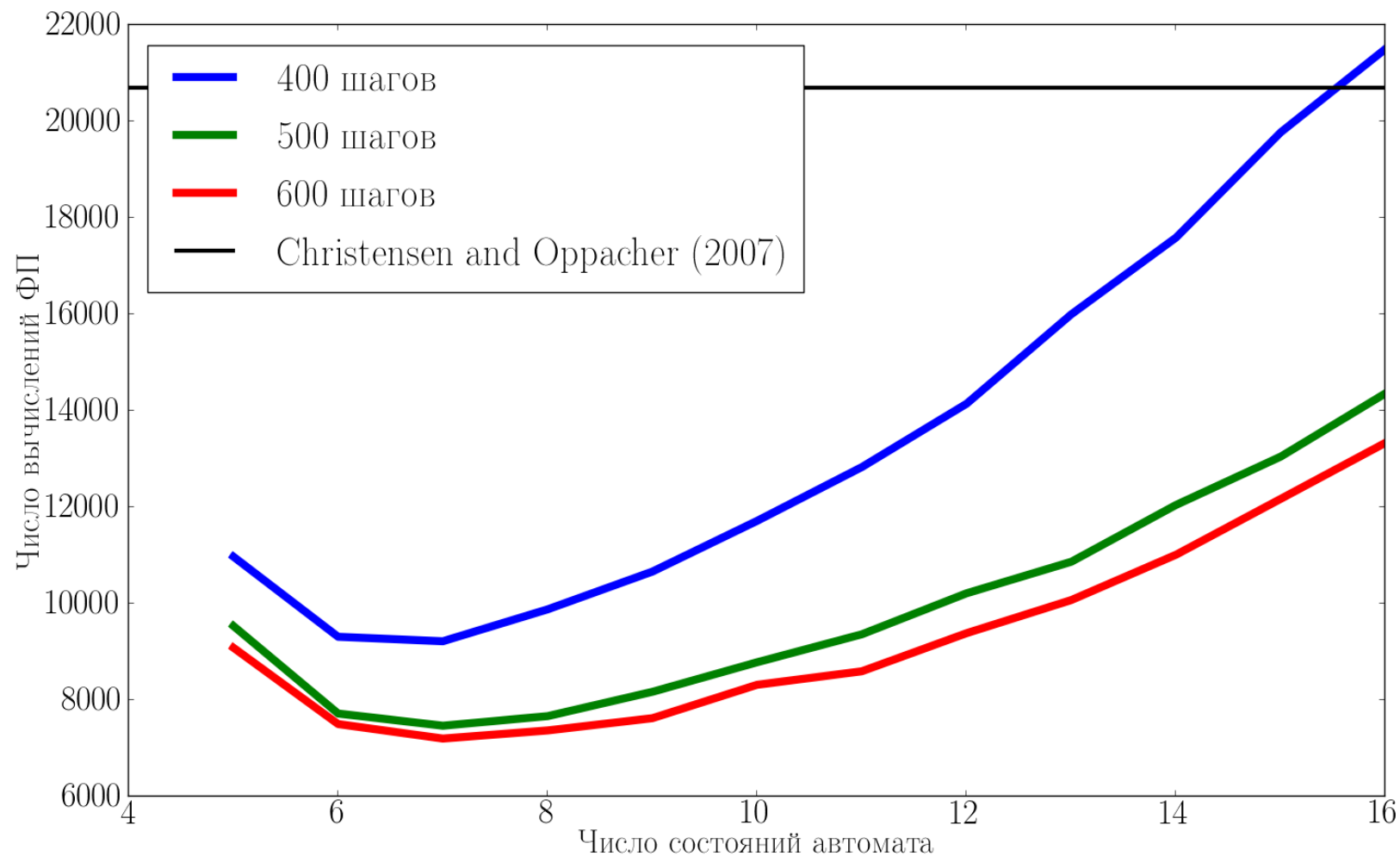


Пример поля

Пример применения: задача «Умный муравей»

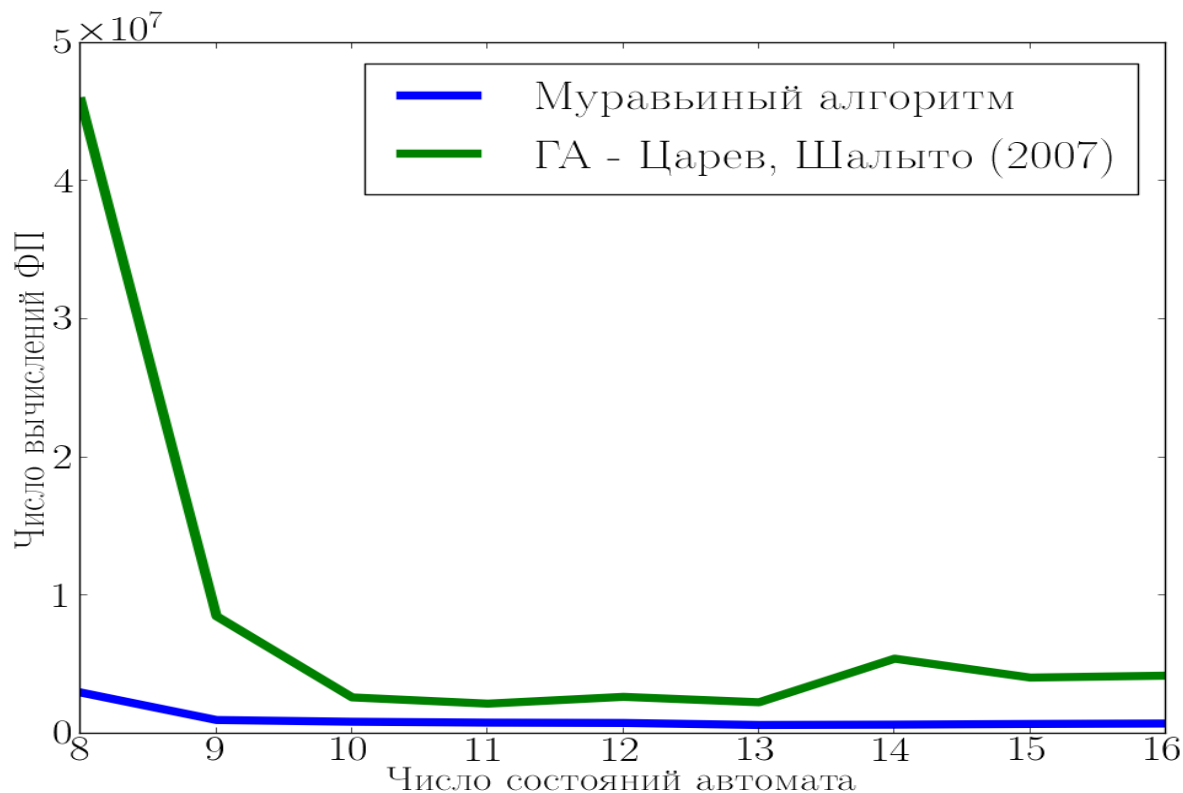
- Два поля:
 - Santa Fe Trail
 - John Muir Trail
- Сравнение с работами:
 - Christensen, Oppacher (2007)
 - Царев, Шалыто (2007)

Santa Fe trail



John Muir Trail (Царев и Шалыто, 2007): 200 ходов

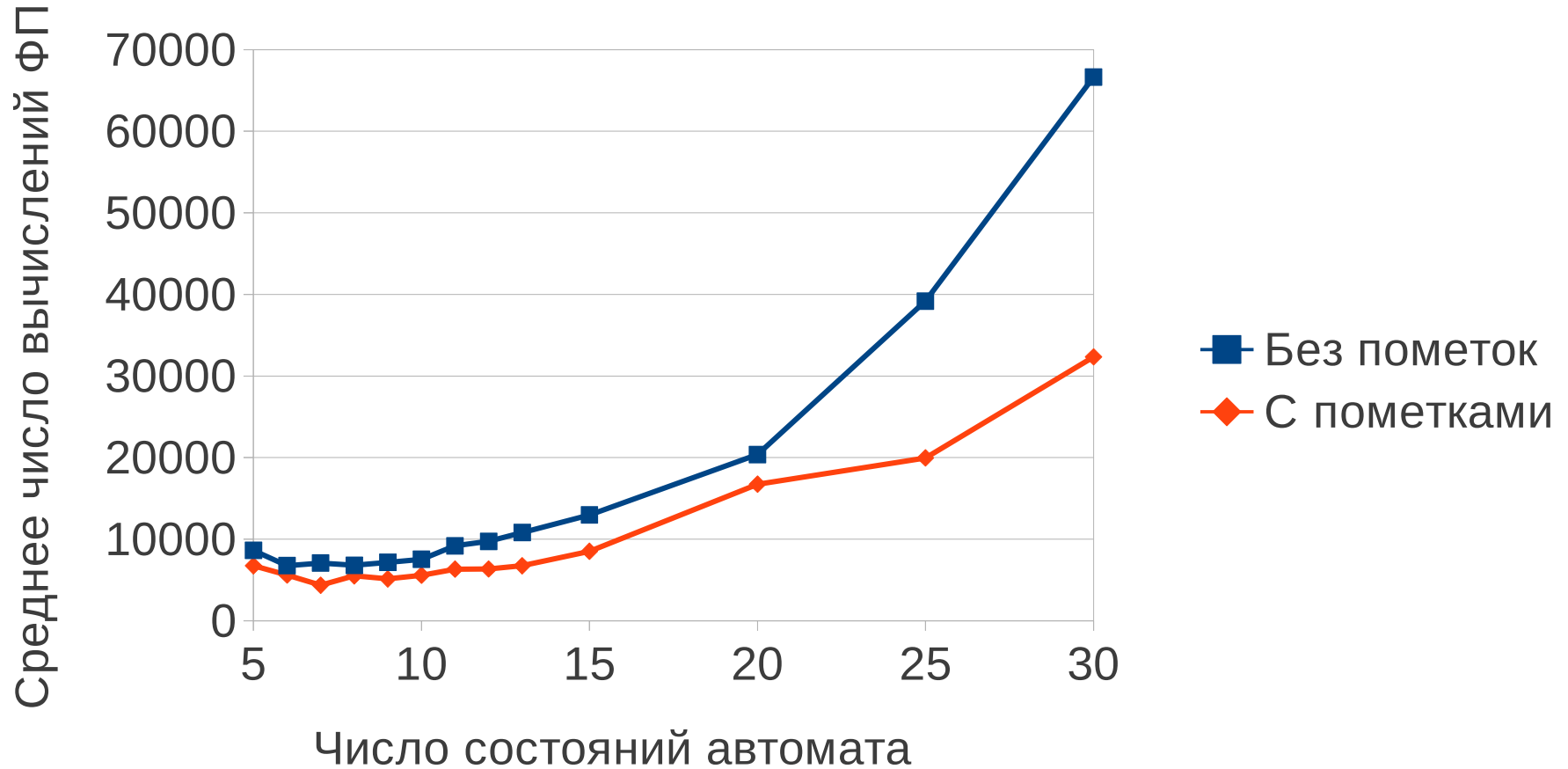
Для
автоматов с
семью
состояниями
— в 30 раз
быстрее



Учет использованных переходов автомата

- При вычислении ФП пометим переходы, которые делал автомат
- Если мутация изменяет неиспользуемый переход — не будем пересчитывать ФП

Santa Fe Trail с учетом использованных переходов (600 ходов)



Результаты

1. Разработан метод построения автоматов на основе муравьиного алгоритма
2. Для задачи об «Умном муравье» метод эффективнее других алгоритмов