

Метод построения конечных автоматов на основе муравьиного алгоритма

Магистерская диссертация

Чивилихин Даниил Сергеевич, гр. 6539

Научный руководитель: канд. техн. наук

Царев Федор Николаевич

22.06.2013

НИУ ИТМО, кафедра Компьютерных технологий

Актуальность

- Автоматное программирование
 - Поведение программ задается конечными автоматами
 - возможность верификации
- Строить автоматы вручную трудно
- Автоматы строятся методами поисковой оптимизации

Новизна

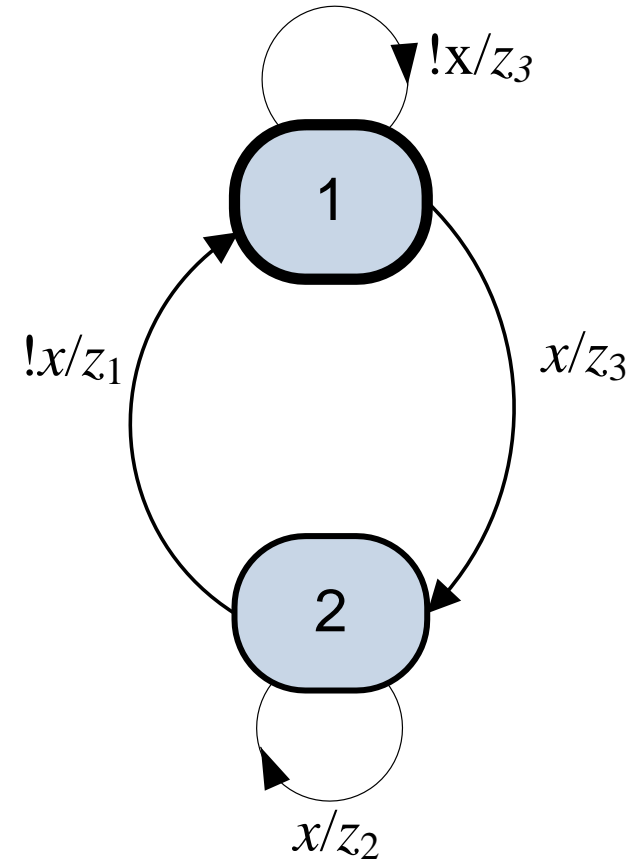
- Муравьиные алгоритмы относятся к методам роевого интеллекта (*swarm intelligence*)
- Ни один из методов роевого интеллекта до сих пор не применялся для построения автоматов

Цель работы

- Разработать метод автоматизированного построения конечных автоматов, основанный на муравьином алгоритме

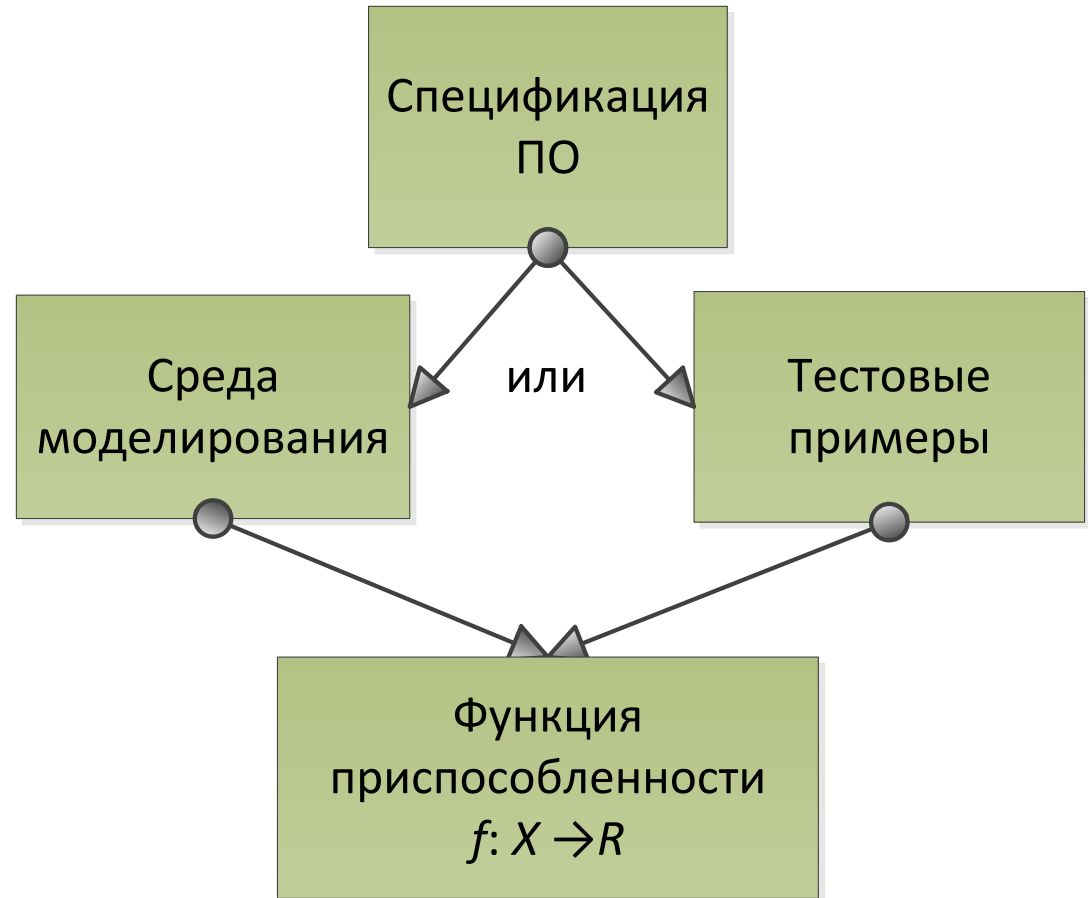
Конечный автомат

- S – множество состояний;
- $s_0 \in S$ – начальное состояние;
- Σ – множество входных событий;
- Δ – множество выходных воздействий;
- $\delta: S \times \Sigma \rightarrow S$ – функция переходов;
- $\lambda: S \times \Sigma \rightarrow \Delta$ – функция действий.



Автоматизированное построение конечных автоматов

- N – число состояний;
- Σ – множество входных событий;
- Δ – множество выходных воздействий;
- $X = (N, \Sigma, \Delta)$



Классический муравьиный алгоритм

- Комбинаторные задачи сводятся к поиску оптимального пути в некотором графе
- Вершины графа — переходы автомата:
 - $\langle i \in S, j \in S, e \in \Sigma, a \in \Delta \rangle$
- Граф – полный
- Муравьи добавляют переходы в автомат, перемещаясь по вершинам графа

Предлагаемый метод построения автоматов

- Альтернативное сведение к оптимизации на графе
 - **Муравьиный алгоритм** нового типа
 - Классические муравьиные алгоритмы неэффективны для решения задачи
- 

Отличие от классических муравьиных алгоритмов: граф

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

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

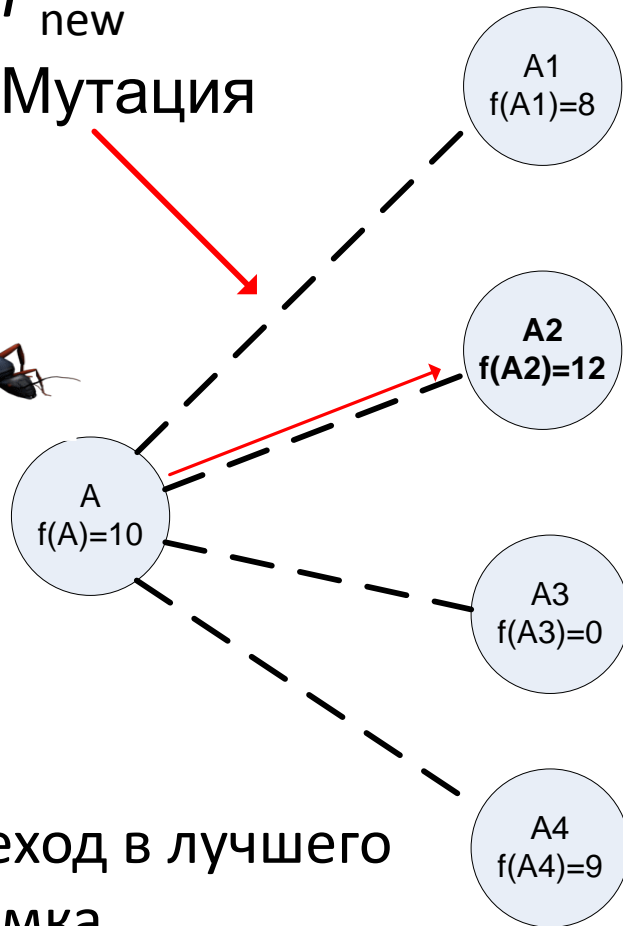
Построение решений муравьями

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

Выбор следующей вершины

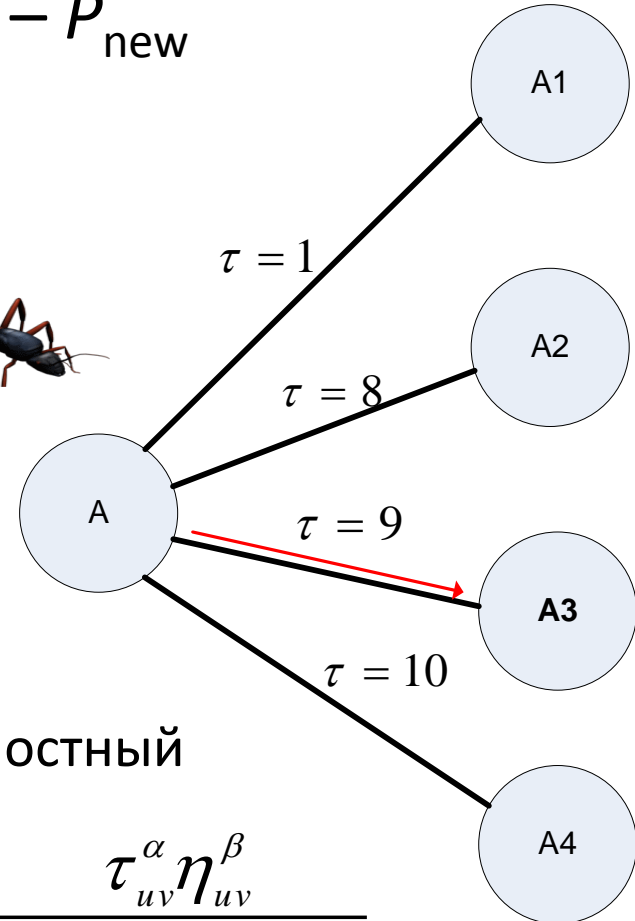
$$P = P_{\text{new}}$$

Мутация



Переход в лучшего потомка

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



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:

$$\tau_{uv} = (1 - \rho)\tau_{uv} + \tau_{uv}^{best}$$

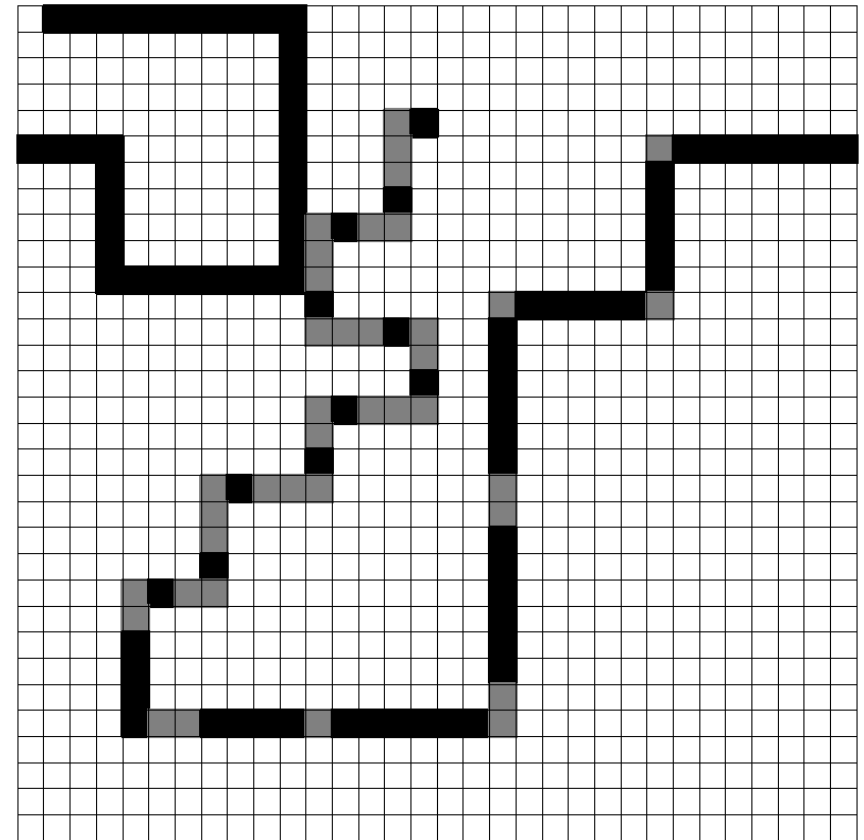
- $\rho \in [0, 1]$ – скорость испарения феромона

Отличие от классических муравьиных алгоритмов

- Граф
- Алгоритм работы муравья
- Критерий остановки муравья

Пример применения: задача «Умный муравей»

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



Пример поля

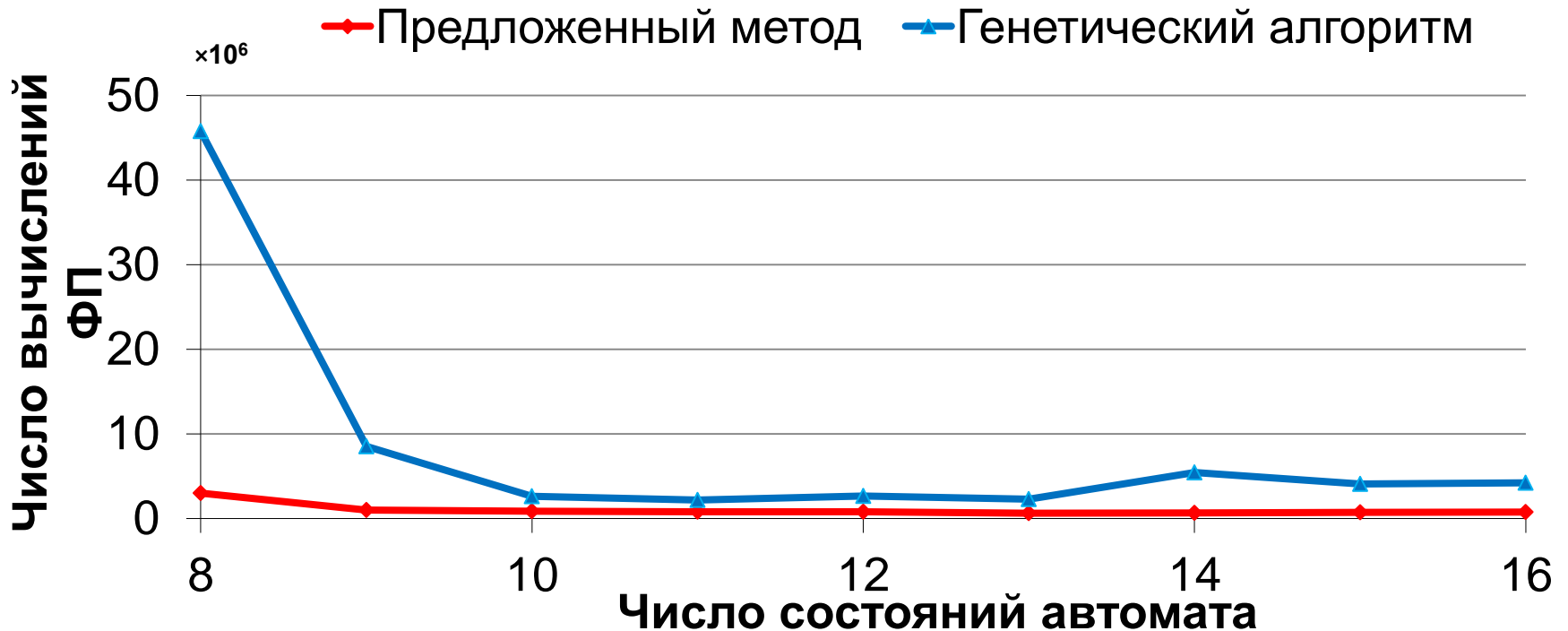
Пример применения: задача «Умный муравей»

- Два поля:
 - Santa Fe Trail
 - John Muir Trail
- Сравнение:
 - **Классический муравьиный алгоритм**
 - Christensen, Oppacher (2007)
 - **Царев, Шалыто (2007)**

Классический муравьиный алгоритм

	Классический муравьиный алгоритм		Предложенный алгоритм	
Число состояний	Доля удач, %	Время, сек	Доля удач, %	Время, сек
5	18	18.09	87	0.65
10	10	218.49	91	0.5

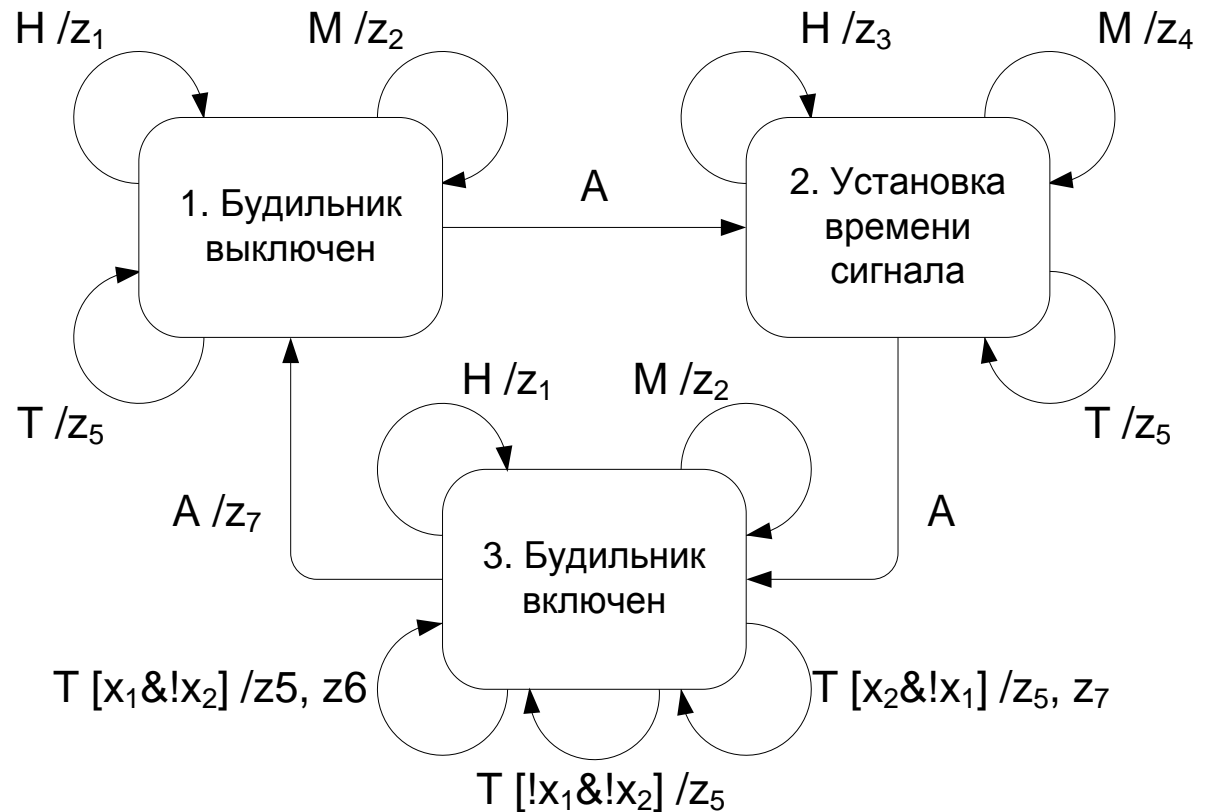
John Muir Trail (Царев и Шалыто, 2007): 200 ходов



- Для автоматов с семью состояниями — в 30 раз быстрее

Построение управляющих автоматов по тестовым примерам: часы с будильником

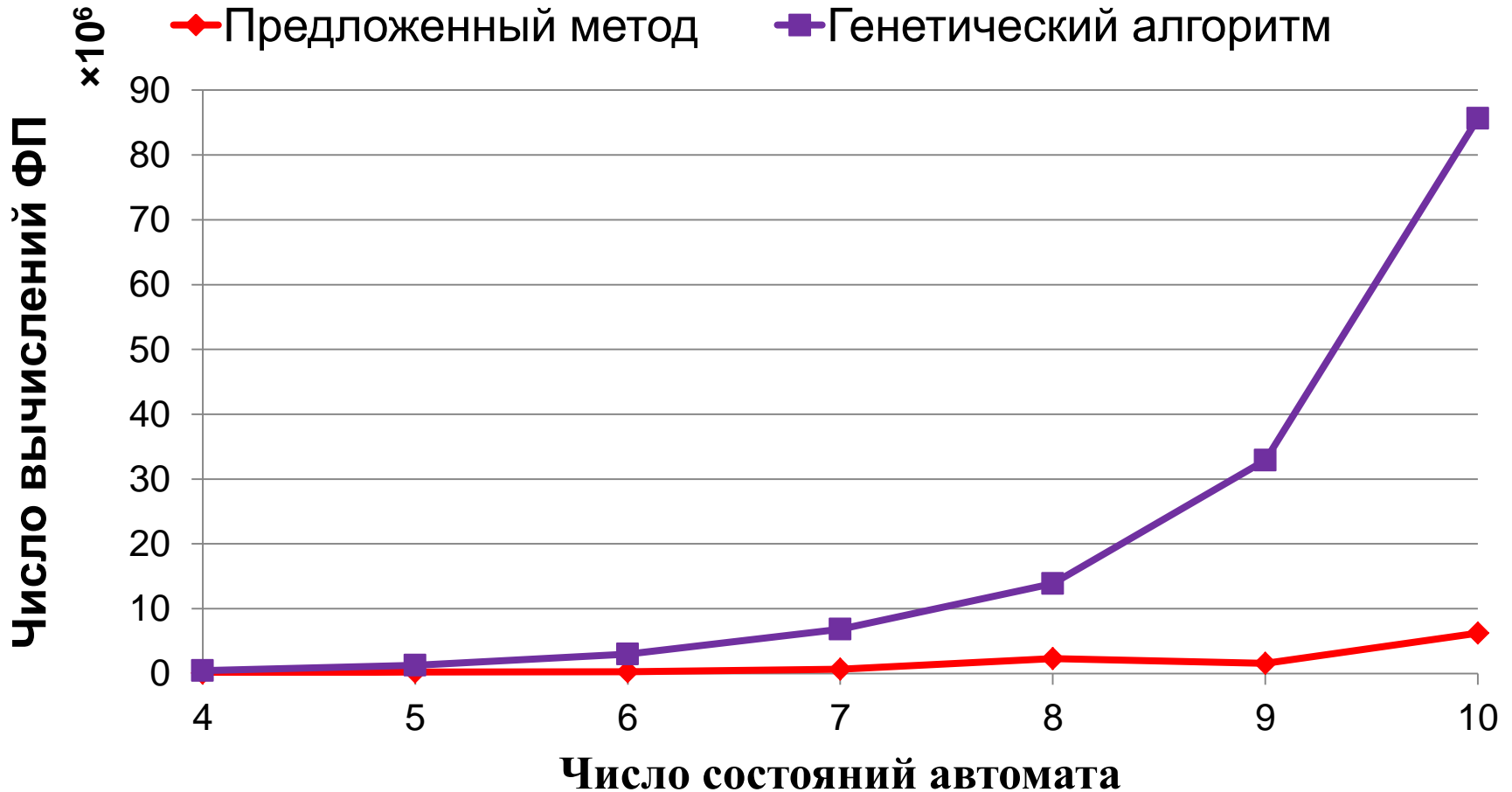
- 38 тестов
- $Len_{in} = 242$
- $Len_{out} = 195$



Построение управляющих автоматов по тестовым примерам: часы с будильником

Метод поисковой оптимизации	Min	Max	Mean
ГА-1	855390	38882588	5805943
RMHC	1150	9592213	1423983
ЭС	1506	9161811	3447390
ГА-2	32830	599022	117977
ГА-2+ЭС	26740	188509	53706
Предложенный метод на основе графа мутаций	2987	211378	34201

Построение управляющих автоматов по тестовым примерам: случайные автоматы (4-10 состояний)



Результаты

- Разработан метод построения автоматов, основанный на муравьином алгоритме нового типа
- Для рассмотренных задач метод в несколько раз эффективнее известных подходов
- Метод применялся для построения автопилота в бакалаврской работе И.Бужинского

ОСНОВНЫЕ ПУБЛИКАЦИИ

1. **Chivilikhin D., Ulyantsev V., Tsarev F.** Test-Based Extended Finite-State Machines Induction with Evolutionary Algorithms and Ant Colony Optimization // Proceedings of the 2012 GECCO Conference Companion on Genetic and Evolutionary Computation. NY.: ACM. 2012, pp. 603–606.
2. **Chivilikhin D., Ulyantsev V.** Learning Finite-State Machines with Ant Colony Optimization // Lecture Notes in Computer Science, 2012, Volume 7461/2012, pp. 268-275.
3. **Chivilikhin D., Ulyantsev V.** MuACOsm - A New Mutation-Based Ant Colony Optimization Algorithm for Learning Finite-State Machines // To appear in Proceedings of the 2013 Genetic and Evolutionary Computation Conference
4. **Chivilikhin D., Ulyantsev V., Shalyto A.** Solving Five Instances of the Artificial Ant Problem with Ant Colony Optimization // To appear in Proceedings of the 2013 IFAC Conference on Manufacturing Modelling, Management and Control
5. **Чивилихин Д.С., Ульянцев В.И.** Метод построения управляющих автоматов на основе муравьиных алгоритмов // Научно-технический вестник информационных технологий, механики и оптики. 2012. №6(82), с. 72-76.

Спасибо за внимание!