

Университет ИТМО

Чивилихин Даниил Сергеевич

Генерация конечных автоматов на основе муравьиных алгоритмов

Диссертация на соискание ученой степени
кандидата технических наук

Специальность 05.13.11.

„Математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей“

Научный руководитель — доктор технических наук, профессор
Шалыто Анатолий Абрамович

Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (1/10)

Актуальность

- ▶ **Автоматное программирование** – парадигма программирования, в которой поведение программ описывается **конечными автоматами** [Поликарпова Н.И., Шалыто А.А. *Автоматное программирование* (2009)]
 - ▶ Применяется в средах UNIMOD (ИТМО), MATLAB/Stateflow, IBM Rhapsody
 - ▶ В средах разработки приложений промышленной автоматике по международным стандартам IEC 61131 и IEC 61499
- ▶ **Достоинства**
 - ▶ Наглядность описания поведения программ
 - ▶ Автоматическая генерация кода
 - ▶ Высокий уровень автоматизации верификации с помощью метода *model checking*
- ▶ **Построение автоматов**
 - ▶ Вручную: долго, сложно, не всегда возможно
 - ▶ Применяются методы автоматической генерации автоматов
- ▶ **Автоматическая генерация автоматов**
 - ▶ На основе моделирования [Spears (2000), Царев (2007), Поликарпова (2010)]
 - ▶ На основе сценариев (тестов) [Lang (1998), Heule (2010), Царев (2012)]
 - ▶ На основе темпоральных формул [Jobstmann (2006), Ehlers (2012), Cheng (2014)]
 - ▶ Сценарии + темпоральные формулы [Царев (2012), Егоров (2013)]
 - ▶ рассматривается в диссертации
- ▶ **Задача генерации автоматов по сценариям и темпоральным формулам**
 - ▶ Полна в сложностном классе 2-EXPTIME [Rosner, 1992]
 - ▶ Существуют как точные, так и другие решения

Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (2/10)

Актуальность. Методы генерации конечных автоматов с учетом темпоральных формул

- ▶ „Точные“ методы генерации автоматов по темпоральным формулам (AE-парадигма)
 - ▶ *Jobstmann B., Bloem R. Optimizations for LTL Synthesis (2006)*
 - ▶ *Ehlers R. Symbolic bounded synthesis (2012)*
 - ▶ *Cheng C.-H. et al. G4LTL-ST: Automatic Generation of PLC Programs (2014)*
 - ▶ **Обладают двойной экспоненциальной сложностью**
 - ▶ **Поэтому долго работают**
- ▶ Метаэвристические алгоритмы для генерации автоматов по сценариям и темпоральным формулам
 - ▶ *Царев Ф. Н. Методы построения конечных автоматов на основе эволюционных алгоритмов (2012)*
 - ▶ *Егоров К. В. Генерация управляющих автоматов на основе генетического программирования и верификации (2013)*
 - ▶ **Недостаточно эффективны** из-за типа применяемых алгоритмов

Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (3/10)

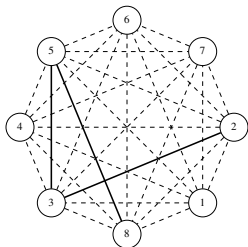
Актуальность

- ▶ Задачу генерации автоматов можно рассматривать как задачу оптимизации
- ▶ **Метаэвристики** – „эвристики о том, как создавать эвристики“
- ▶ Позволяют находить „хорошие“ решения сложных задач оптимизации за приемлемое время [Blum et al. *Metaheuristics in Combinatorial Optimization* (2003); Курейчик и др. *Биоинспирированные алгоритмы в оптимизации* (2010)]:
 - ▶ Эволюционные и генетические алгоритмы
 - ▶ Метод имитации отжига
 - ▶ Муравьиные алгоритмы [Dorigo et al. *Ant System: optimization by a colony of cooperating agents* (1996)]
 - ▶ **Используются в диссертации**, ранее не применялись для генерации автоматов
- ▶ Метаэвристики работают плохо, когда рядом с оптимальными решениями располагается много субоптимальных
 - ▶ Такая ситуация наблюдается в задачах генерации конечных автоматов
 - ▶ **Муравьиные алгоритмы** в силу своих свойств могут помочь справиться с этой проблемой
- ▶ **Следовательно**, задача разработки эффективных методов генерации конечных автоматов по сценариям и темпоральным формулам на основе муравьиных алгоритмов является **актуальной**.

Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (4/10)

Муравьиные алгоритмы

- ▶ Метаэвристические алгоритмы, предназначенные для решения задач оптимизации на графах
- ▶ Например, задача о коммивояжере
 - ▶ Найти гамильтонов путь минимального веса во взвешенном графе
- ▶ На ребрах графа задаются вспомогательные величины
 - ▶ Значение феромона – вспомогательная величина, используется алгоритмом
 - ▶ Эвристическая информация – оценка локального решения



Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (5/10)

Цель и задачи диссертационной работы

- ▶ Цель работы — развитие существующих подходов к генерации конечных автоматов по сценариям работы и темпоральным формулам за счет применения муравьиных алгоритмов
- ▶ Для достижения цели решаются следующие задачи:
 1. Разработать метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма. **(Глава 2)**
 2. Разработать методы генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов. **(Глава 3)**
 3. Разработать инструментальное средство, реализующее предложенные методы. **(Глава 4)**
 4. Внедрить результаты работы при генерации диаграмм управления выполнением для базисных функциональных блоков стандарта IEC 61499 и в учебный процесс. **(Глава 5)**

Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (6/10)

Научные положения, выносимые на защиту, которые обладают научной новизной

1. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам, основанный на муравьином алгоритме, отличающийся от существующих тем, что в нем используется граф мутаций, предложенный автором

Экспериментально обосновано, что этот метод работает быстрее известных методов на основе генетических алгоритмов и АЕ-парадигмы

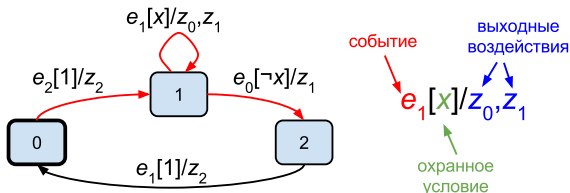
2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам, совмещающий:
 - ▶ параллельный муравьиный алгоритм
 - ▶ точный метод генерации конечных автоматов по сценариям работы на основе сведения к задаче выполнимости булевой формулы

Экспериментально обосновано, что метод работает быстрее параллельного генетического алгоритма и параллельного муравьиного алгоритма

Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (7/10)

Управляющие конечные автоматы

- ▶ Конечный автомат – семерка $\langle E, Y, Z, y_0, \phi, \delta \rangle$, E – множество входных событий, Y – множество состояний, $y_0 \in Y$ – начальное состояние, Z – множество выходных воздействий, $\phi: Y \times E \rightarrow Y$ – функция переходов, $\delta: Y \times E \rightarrow Z^*$ – функция выходов
- ▶ Особенностью **управляющих конечных автоматов** является то, что пометки на переходах могут содержать события и логические условия на входные переменные (охранные условия)

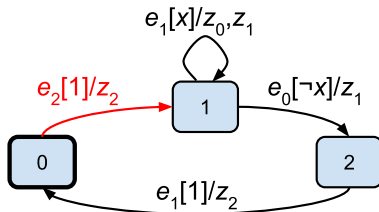


- ▶ **Сценарий работы** – последовательность троек $\langle e \in E, \varphi \in 2^X, O \in Z^* \rangle$
 - ▶ Соответствует пути в автомате
- ▶ Автомат удовлетворяет сценарию $\langle e_2, 1, (z_2) \rangle \langle e_1, x, (z_0, z_1) \rangle \langle e_0, \neg x, (z_1) \rangle$

Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (8/10)

Темпоральные формулы: логика линейного времени (LTL)

- ▶ Предложена Pnueli в 1977
- ▶ Пропозициональные переменные – элементарные высказывания
 - ▶ $\text{wasEvent}(e)$ – выполнен переход, помеченный событием e
 - ▶ $\text{wasAction}(z)$ – выполнен переход, помеченный воздействием z
- ▶ Логические операторы: \wedge, \vee, \neg
- ▶ Темпоральные операторы:
 - ▶ **Globally.** $G(f)$ – f выполняется всегда.
 - ▶ **next.** $X(t)$ – f выполняется в следующем состоянии.
 - ▶ **Future.** $F(f)$ – f выполняется в каком-либо последующем состоянии.
- ▶ Пример: автомат удовлетворяет *LTL*-формуле $G(\text{wasEvent}(e_2) \rightarrow \text{wasAction}(z_2))$



Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (9/10)

Постановка задачи генерации конечных автоматов по сценариям работы и темпоральным формулам

- ▶ Задано
 - ▶ Набор сценариев работы
 - ▶ Набор *LTL*-формул
 - ▶ Число состояний N
- ▶ Требуется: найти автомат с не более чем N состояниями, удовлетворяющий заданным сценариям и *LTL*-формулам

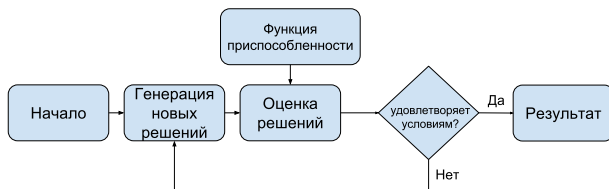
Глава 1. Автоматное программирование, конечные автоматы, методы генерации конечных автоматов (10/10)

Выводы по главе 1

1. Проведен обзор по темам «Автоматное программирование», «Метаэвристические алгоритмы», «Методы генерации конечных автоматов».
2. Существующие методы генерации конечных автоматов с учетом темпоральных формул требуют много времени для нахождения решения даже для простых примеров небольших автоматов.
3. Муравьиные алгоритмы ранее не применялись для генерации конечных автоматов.
4. Показано, что генерация конечных автоматов с помощью классических муравьиных алгоритмов неэффективна.
5. Приведена формальная постановка задачи генерации конечных автоматов по сценариям работы и *LTL*-формулам.
6. Сформулирована цель и задачи диссертационной работы.

Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (1/14)

Метаэвристический подход

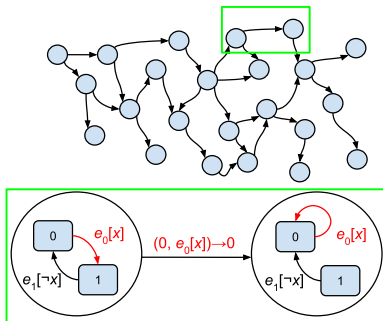


- ▶ Для оценки соответствия решений-кандидатов условиям задачи используется так называемая функция приспособленности (ФП)
- ▶ Используется ФП из диссертаций Ф. Царева (2012) и К. Егорова (2013)

Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (2/14)

Способ представления пространства поиска – граф мутаций

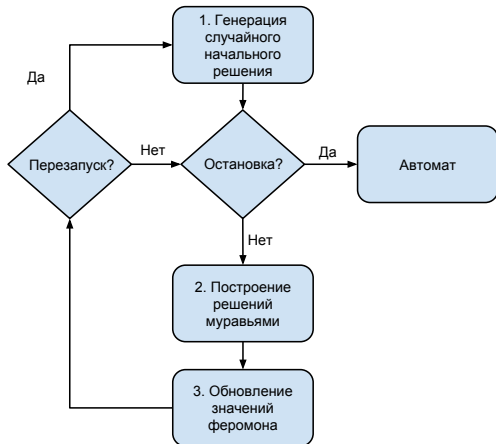
- ▶ Вершины – конечные автоматы
- ▶ Ребра – мутации конечных автоматов (небольшие изменения)
 - ▶ Операторы мутации из диссертаций Ф. Царева (2012) и К. Егорова (2013)
- ▶ На ребре uv заданы две величины
 - ▶ Значение феромона τ_{uv}
 - ▶ Эвристическая информация $\eta_{uv} = \max(\eta_{\min}, F(v) - F(u))$



Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (3/14)

Предлагаемый метод „муравьиный алгоритм с мутациями”
(Mutation-based Ant Colony Optimization)

- ▶ Mutation-based Ant Colony Optimization – MuACO



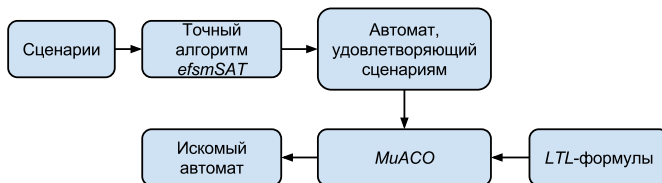
Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (4/14)

2.1. Генерация начального решения

1. Генерация случайного автомата

- ▶ С заданным числом состояний
- ▶ Переход в случайное состояние для каждой комбинации события в охранного условия, встречающейся в сценариях

2. Только по сценариям работы с помощью точного метода на основе решений задачи выполнимости (SAT) [Ulyantsev, 2011]



Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (5/14)

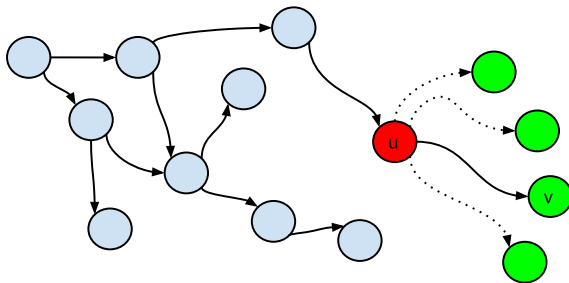
2.2. Построение решений муравьями

- ▶ Используется колония из N_{ants} муравьев
- ▶ Все муравьи стартуют из вершины, ассоциированной с лучшим на данный момент решением
- ▶ Муравей выполняет не более n_{stag} шагов без увеличения значения ФП
- ▶ На каждом шаге муравей переходит в следующую вершину с помощью одного из двух механизмов:
 1. построение новых решений
 2. вероятностный выбор
- ▶ Опишем эти механизмы

Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (6/14)

2.2.1. Построение новых решений

- ▶ Муравей находится в вершине u , ассоциированной с автоматом A
- ▶ Создаются новые ребра и вершины графа мутаций путем выполнения фиксированного числа мутаций N_{mut} автомата A
- ▶ Муравей выбирает **лучшую** из построенных вершин v и переходит в нее

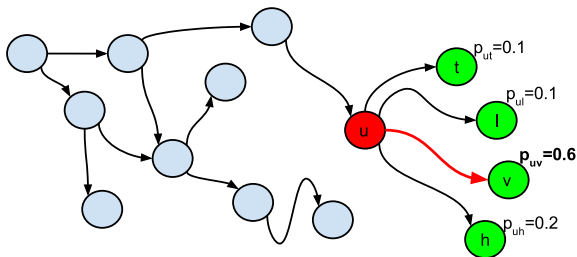


Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (7/14)

2.2.2. Вероятностный выбор

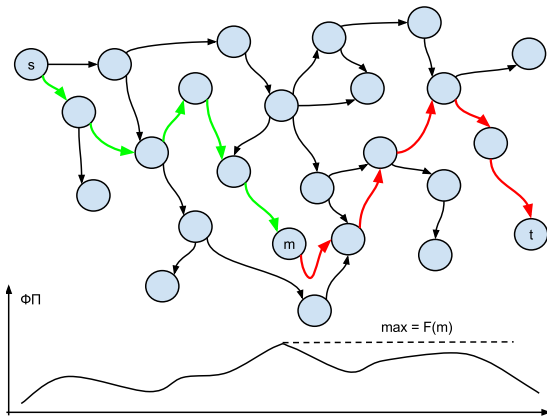
- ▶ Муравей находится в вершине u , ассоциированной с автоматом A
- ▶ Следующая вершина v выбирается из множества N_u вершин, инцидентных вершине u , с вероятностью

$$p_{uv} = \frac{\tau_{uv}^\alpha \cdot \eta_{uv}^\beta}{\sum_{w \in N_u} \tau_{uw}^\alpha \cdot \eta_{uw}^\beta}$$



Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (8/14)

2.3. Обновление значений феромона



- ▶ $\tau_{uv} = \max(\tau_{\min}, (1 - \rho)\tau_{uv} + \tau_{uv}^{\text{best}})$, $\rho \in [0, 1]$ – скорость испарения феромона, τ_{\min} – минимальное разрешенное значение феромона

Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (9/14)

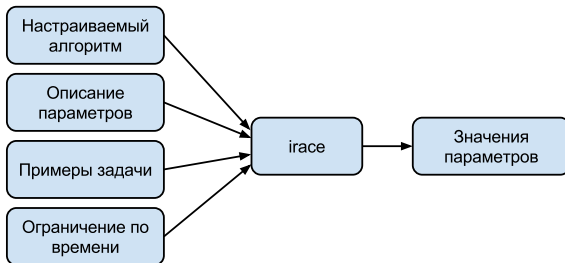
Вычислительные эксперименты

- ▶ Проводится сравнение предложенных методов *MuACO* и *SAT+MuACO* с генетическим алгоритмом и алгоритмами на основе *AE*-парадигмы
- ▶ Методология проведения экспериментов по сравнению алгоритмов
 - ▶ I. Настройка значений параметров алгоритмов
 - ▶ II. Генерация автомата управления дверьми лифта
 - ▶ III. Генерация автоматов по случайно сгенерированным входным данным

Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (10/14)

I. Настройка значений параметров алгоритмов

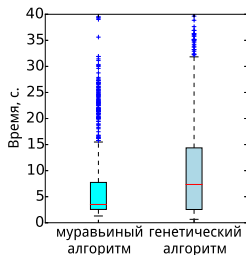
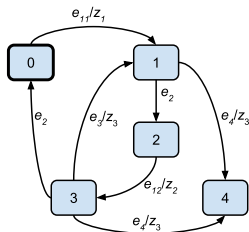
- ▶ Рассматриваемые алгоритмы имеют множество параметров
 - ▶ Как выбрать их значения?
- ▶ Используется программное средство *irace* (<http://iridia.ulb.ac.be/irace/>), позволяющее выбрать значения параметров *автоматически*



Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (11/14)

II. Генерация автомата управления дверьми лифта: *MuACO* vs генетический алгоритм

- ▶ Пять входных событий, три выходных воздействия, девять сценариев, 13 *LTL*-формул
 - ▶ 1000 запусков каждого алгоритма
 - ▶ медиана времени работы *MuACO* равна 3,5 с, генетического – 7,3 с
 - ▶ значение *p-value* (тест Уилкоксона) равно $3,93 \cdot 10^{-13}$



- ▶ **Вывод:** для этого примера *MuACO* в среднем работает быстрее генетического алгоритма

Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (12/14)

II. Генерация автомата управления дверьми лифта: *MuACO* vs *AE*-парадигма

- ▶ Программные средства на основе *AE*-парадигмы
 - ▶ *lily* [Jobstmann et al, 2006]
 - ▶ *unbeast* [Ehlers, 2015]
 - ▶ *G4LTL_ST* [Cheng et al, 2014]

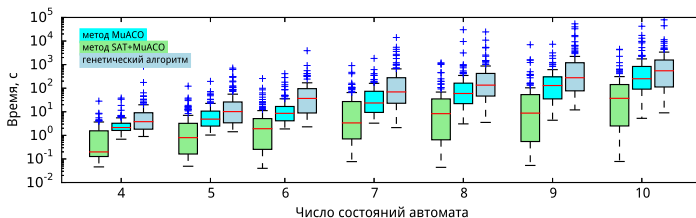
	Программное средство			
	<i>MuACO</i>	<i>lily</i>	<i>unbeast</i>	<i>G4LTL_ST</i>
Число состояний	5	15	11	30
Время, с	3,5	453	6	9

- ▶ **Вывод:** *MuACO* работает быстрее методов на основе *AE*-парадигмы и генерирует решения с меньшим числом состояний

Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (13/14)

III. Генерация автоматов по случайно сгенерированным входным данным

- ▶ Сравниваются методы *MuACO*, *SAT+MuACO* и генетический алгоритм
- ▶ Число состояний N_{states} – от четырех до десяти, два входных события, одна входная переменная, два выходных воздействия
- ▶ По 50 запусков каждого алгоритма для каждого значения N_{states}



- ▶ *MuACO* примерно в два раза быстрее генетического алгоритма
- ▶ *SAT+MuACO* примерно в 4–14 раз быстрее, чем *MuACO*
- ▶ Максимальное значение *p-value* (тест Уилкоксона) равно $5 \cdot 10^{-4}$

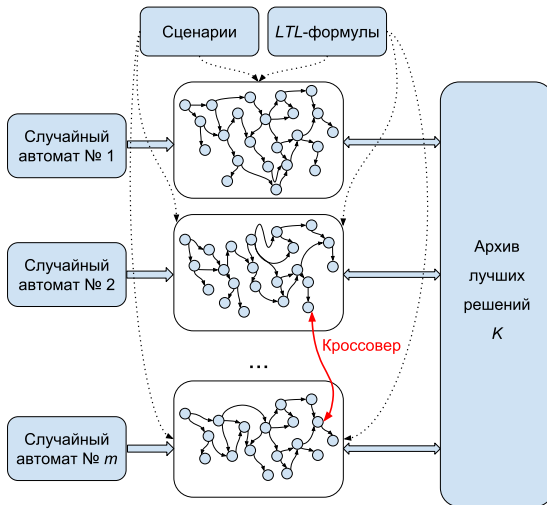
Глава 2. Метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма (14/14)

Выводы по главе 2

- ▶ Предложен метод генерации конечных автоматов по сценариям работы и темпоральным формулам на основе муравьиного алгоритма
- ▶ Проведены вычислительные эксперименты – предложенный метод превосходит по эффективности известные методы на основе генетического алгоритма и *AE*-парадигмы
- ▶ Результаты, описанные в данной главе, опубликованы в трудах международной конференции „Genetic and Evolutionary Computation Conference“ и в журналах из перечня ВАК
- ▶ Предложенный метод также применялся для решения задач, в которых функция приспособленности вычисляется на основе моделирования

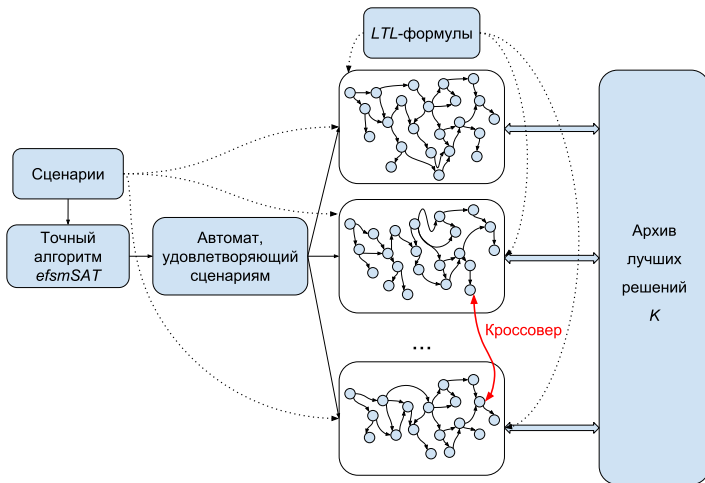
Глава 3. Методы генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов (1/5)

1. Параллельный муравьиный алгоритм $pMuACO$ (parallel MuACO)



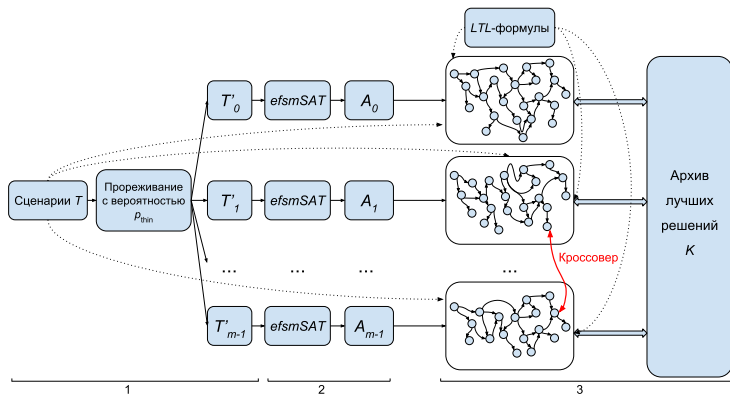
Глава 3. Методы генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов (2/5)

2. Параллельный муравьиный алгоритм *psMuACO* (parallel SAT MuACO)



Глава 3. Методы генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов (3/5)

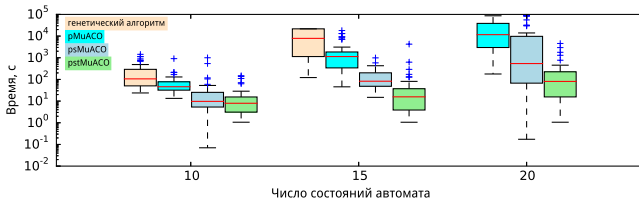
3. Предлагаемый метод генерации автоматов *pstMuACO* (parallel SAT thin-out MuACO)



Глава 3. Методы генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов (4/5)

Вычислительные эксперименты

- Сравниваются методы *pMuACO*, *psMuACO*, *pstMuACO* и параллельный генетический алгоритм



Значения *p-value* (тест Уилкоксона)

Число состояний	10	15	20
<i>pMuACO</i> vs. параллельный ГА	$7,6 \cdot 10^{-6}$	$1,42 \cdot 10^{-6}$	—
<i>psMuACO</i> vs. <i>pMuACO</i>	$1,035 \cdot 10^{-5}$	$2,2 \cdot 10^{-5}$	$2,5 \cdot 10^{-3}$
<i>pstMuACO</i> vs. <i>psMuACO</i>	0,03	$3,264 \cdot 10^{-5}$	$7,8 \cdot 10^{-5}$

Вывод: *pstMuACO* в среднем более, чем в 40 раз быстрее алгоритма *pMuACO* и в 11 раз быстрее *psMuACO*

Глава 3. Методы генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов (5/5)

Выводы

1. Предложен метод $pMuACO$ на основе параллельного запуска нескольких взаимодействующих алгоритмов $MuACO$.
2. Предложен метод $psMuACO$, в котором начальное решение для всех потоков генерируется с помощью точного алгоритма $efsmSAT$ построения автоматов по сценариям работы.
3. Предложен метод генерации конечных автоматов по сценариям работы и темпоральным формулам $pstMuACO$, совмещающий параллельный муравьиный алгоритм $pMuACO$, точный алгоритм на основе решения SAT и процедуру прореживания сценариев. Показано, что метод работает быстрее, чем $pMuACO$ и $psMuACO$.
4. Часть описанных в данной главе результатов опубликована в трудах международной конференции „International Student Workshop on Bioinspired Optimization Methods and their Applications“ .
5. Статья по остальным результатам данной главы принята к печати в журнал „Автоматика и телемеханика“: Чивилихин Д.С., Ульяновцев В.И., Шалыто А.А. Модифицированный муравьиный алгоритм для построения конечных автоматов по сценариям работы и темпоральным формулам.

Глава 4. Инструментальное средство и библиотека для генерации конечных автоматов (1/1)

Инструментальное средство для генерации конечных автоматов

- ▶ Разработанные методы реализованы на языке *Java* в виде программного средства *tuaco.jar*
 - ▶ Исходный код размещен в открытом доступе в сети Интернет по адресу: https://bitbucket.org/chivilikhin_daniil/aco-for-automata
 - ▶ Входные данные – сценарии и темпоральные формулы в текстовом формате
 - ▶ Выходные данные – граф переходов конечного автомата, записанный в формате *DOT* (www.graphviz.org)
- ▶ Программное средство может быть использовано в качестве библиотеки для решения других задач
 - ▶ Например, для решения задачи, описанной в главе 5
- ▶ Получено два свидетельства о регистрации программы для ЭВМ
 - ▶ № 2013661249 от 3 декабря 2013 года «Программное средство, реализующее муравьиный алгоритм для построения конечных автоматов», авторы Чивилихин Д.С., Ульяновцев В.И.
 - ▶ № 2015610291 от 12 января 2015 года «Библиотека параллельных муравьиных алгоритмов для построения управляющих конечных автоматов», авторы Чивилихин Д.С., Ульяновцев В.И.

Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (1/9)

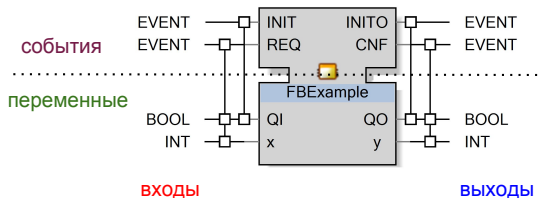
Стандарт разработки приложений промышленной автоматки IEC 61499

- ▶ IEC 61499 – международный стандарт разработки приложений промышленной информатики
- ▶ Программы представляются в виде **функциональных блоков (ФБ)**
 - ▶ Элементарный компонент приложения
 - ▶ Обрабатывает данные и события
 - ▶ ФБ рекомендовано описывать конечными автоматами

Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (2/9)

Функциональный блок (ФБ)

- ▶ характеризуется *интерфейсом*: определяет входные/выходные события и входные/выходные переменные.
- ▶ переменные могут быть логическими, целочисленными, вещественными, etc.

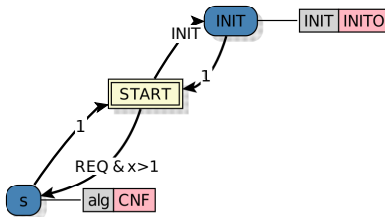


- ▶ *Базисный* ФБ, помимо интерфейса, задается с помощью конечного автомата Мура специального вида, называемого *Execution Control Chart* (диаграмма управления выполнением, ДУВ)
- ▶ *Составной* ФБ задается сетью блоков, каждый из которых может быть как базисным, так и составным.

Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (3/9)

Диаграмма управления выполнением (ДУВ) базисного ФБ

- ▶ начальное состояние
- ▶ переход в новое состояние в случае активации *охранного условия* – булевой формулы от входных, выходных, внутренних переменных
- ▶ состояние может содержать несколько действий
 - ▶ выходное событие
 - ▶ запуск *алгоритма* – изменение значений выходных переменных

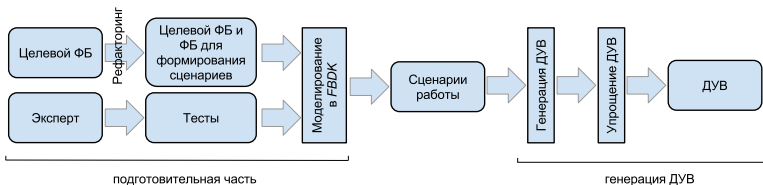


Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (4/9)

Практическая задача

- ▶ Восстановить диаграмму управления выполнением базисного ФБ по примерам поведения
 - ▶ Миграция приложений из стандарта IEC 61131
 - ▶ исходный код утерян
 - ▶ Нет инженеров, способных быстро в нем разобраться

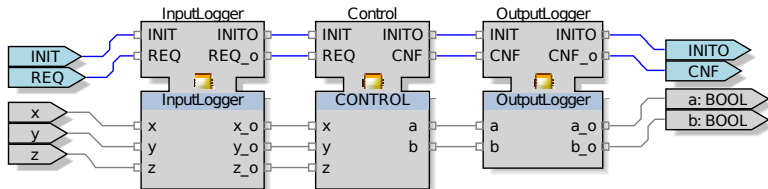
Схема предлагаемого подхода



Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (5/9)

Подготовительная часть

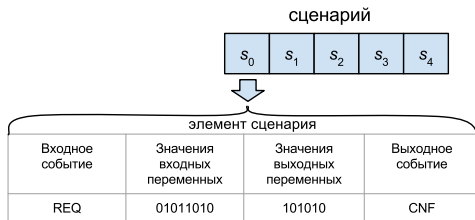
- ▶ Создание тестов: вручную
- ▶ Рефакторинг сети функциональных блоков: автоматически
 - ▶ Целевой блок **Control** дополняется блоками **InputLogger** и **OutputLogger**
 - ▶ InputLogger – сохраняет входные события и значения входных переменных
 - ▶ OutputLogger – сохраняет выходные события и значения выходных переменных



Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (6/9)

Формирование сценариев работы

- ▶ Сценарий работы функционального блока



- ▶ Формирование сценариев работы путем моделирования тестов в среде *FBDK* (<http://www.holobloc.com/doc/fbdk/>) разработки приложений IEC 61499



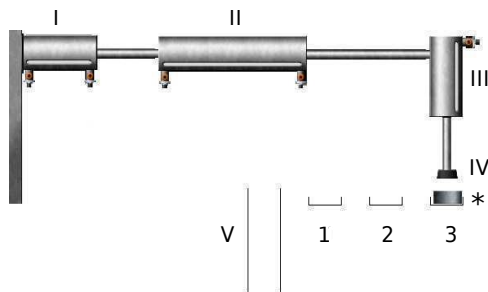
Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (7/9)

Генерация диаграммы управления выполнением, удовлетворяющей сценариям

- ▶ Генерация диаграммы, удовлетворяющей сценариям, с помощью предложенного в главе 3 алгоритма *pMuACO*
- ▶ Разработаны
 - ▶ Способ представления диаграммы с учетом булевых формул произвольного вида
 - ▶ Операторы мутации
 - ▶ Функция приспособленности

Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (8/9)

Пример работы: *Pick-and-Place* манипулятор

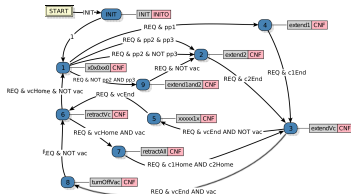


- ▶ Цилиндры (I, II, III) перемещают присоску (IV), захватывающую деталь (*) во входных корзинах (1, 2, 3)
- ▶ Тест – последовательность появления деталей в корзинах (1, 2, 3)

Глава 5. Внедрение результатов работы при генерации автоматной логики для базисных функциональных блоков стандарта IEC 61499 (9/9)

Пример работы: *Pick-and-Place* манипулятор

- ▶ Решение находится в среднем за 4,5 часа
- ▶ Один из сгенерированных автоматов



- ▶ Все решения были проверены путем моделирования тестов в среде *FBDK*
- ▶ По результатам главы 5 опубликовано две статьи в трудах конференций IEEE
 - ▶ Chivilikhin D., Shalyto A., Patil S., Vyatkin V. Reconstruction of Function Block Logic using Metaheuristic Algorithm: Initial Explorations / In Proceedings of the 13th IEEE International Conference on Industrial Informatics (INDIN'15)
 - ▶ Chivilikhin D., Shalyto A., Vyatkin V. Inferring Automata Logic From Manual Control Scenarios: Implementation in Function Blocks / In Proceedings of the 13th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA'15)

Результаты работы

1. Предложен метод *MuACO* генерации конечных автоматов по сценариям работы и темпоральным формулам, основанный на муравьином алгоритме с предложенным автором графом мутаций. Показано, что метод работает быстрее известных методов на основе генетических алгоритмов и АЕ-парадигмы.
2. Предложены методы *pMuACO*, *psMuACO* и *pstMuACO* генерации конечных автоматов по сценариям работы и темпоральным формулам на основе параллельных муравьиных алгоритмов. Последний из них, *pstMuACO*, совмещает параллельный муравьиный алгоритм *pMuACO*, точный метод *efsmSAT* генерации конечных автоматов по сценариям работы на основе сведения к задаче выполнимости булевой формулы и процедуру прореживания сценариев. Показано, что метод *pstMuACO* работает быстрее параллельного генетического алгоритма, а также методов *pMuACO* и *psMuACO*.
3. Разработано программное средство *tuaco.jar*, реализующее предложенные методы, исходный код которого размещен в открытом доступе в сети Интернет. Средство позволяет генерировать автоматы по сценариям работы и темпоральным формулам, а также может быть использовано для решения других задач генерации автоматов.
4. Результаты диссертации:
 - ▶ внедрены в Технологическом Университете Лулео (Швеция) при генерации диаграмм управления выполнением для базисных функциональных блоков стандарта *IEC 61499* имеется акт внедрения
 - ▶ использованы в учебном процессе на кафедре „Компьютерные технологии“ Университета ИТМО в рамках курсов «Теория автоматов и программирование» и „Генетическое программирование“, имеется акт использования

Заключение (2/3)

Результаты работы

- ▶ 18 публикаций, из них:
 - ▶ три статьи в журналах ВАК
 - ▶ семь публикаций в изданиях, индексируемых в Web of Science и Scopus одновременно
 - ▶ 11 публикаций в изданиях, индексируемых только в Scopus
- ▶ Получено два свидетельства о регистрации программ для ЭВМ
 - ▶ № 2013661249 от 3 декабря 2013 года «Программное средство, реализующее муравьиный алгоритм для построения конечных автоматов», авторы Чивилихин Д.С., Ульянов В.И.
 - ▶ № 2015610291 от 12 января 2015 года «Библиотека параллельных муравьиных алгоритмов для построения управляющих конечных автоматов», авторы Чивилихин Д.С., Ульянов В.И.



Апробация

- ▶ сделано 15 докладов на конференциях
- ▶ четыре доклада на российских конференциях
 - ▶ Всероссийская конференция по проблемам информатики СПИСОК (Санкт-Петербург, 2012, 2013 гг.)
 - ▶ Международная конференция „Интегрированные модели и мягкие вычисления в искусственном интеллекте“ (Коломна, 2013 г.)
 - ▶ Всероссийская конференция „Всероссийское совещание по проблемам управления“ (Москва, 2014 г.)
- ▶ 11 докладов на зарубежных конференциях
 - ▶ Международная конференция „Genetic and Evolutionary Computation Conference“ (США, Филадельфия, 2011 г., Нидерланды, Амстердам, 2013 г., Канада, Ванкувер, 2014 г.)
 - ▶ Международная конференция „International Conference on Swarm Intelligence“ (Бельгия, Брюссель, 2012 г.)
 - ▶ Международная конференция „BRICS countries Congress on Computational Intelligence“ (Бразилия, Ресифи, 2013 г.)
 - ▶ Международная конференция „IFAC Conference on Manufacturing Modelling, Management, and Control“ (Санкт-Петербург, 2013 г.)
 - ▶ Международная конференция „International Conference on Machine Learning and Applications“ (США, Майами, 2013 г., США, Детройт, 2014 г.)
 - ▶ Международная конференция „International Student Workshop on Bioinspired Optimization Methods and their Applications“ (Словения, Любляна, 2014 г.)
 - ▶ Международная конференция „IEEE International Conference on Industrial Informatics“ (Великобритания, Кембридж, 2015 г.);
 - ▶ Международный симпозиум „IEEE International Symposium on Parallel and Distributed Processing with Applications“ (Финляндия, Хельсинки, 2015 г.)

Спасибо за внимание!

Университет ИТМО

Чивилихин Даниил Сергеевич

Генерация конечных автоматов на основе муравьиных алгоритмов

Диссертация на соискание ученой степени
кандидата технических наук

Специальность 05.13.11.

„Математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей“

Научный руководитель — доктор технических наук, профессор
Шалыто Анатолий Абрамович

Расширенное экспериментальное исследование

- ▶ Сравниваются методы
 1. *MuACO*, *SAT+MuACO*, генетический алгоритм
 2. *pMuACO*, *psMuACO*, *pstMuACO*, параллельный генетический алгоритм
- ▶ Отличие от основных экспериментов диссертации
 - ▶ Каждый метод запускается на каждом примере задачи несколько раз (M)
- ▶ Методики сравнения алгоритмов
 1. Доверительные интервалы отношений времен работы алгоритмов – рекомендована оппонентом
 2. Проверка гипотез о средних временах работы алгоритмов
 3. Диаграммы логарифмов времен работы

1. Доверительные интервалы отношений времен работы алгоритмов

- ▶ В данной методике предполагается, что распределение логарифмов отношений времен работы алгоритмов – нормальное
- ▶ К сожалению гипотеза не выполняется для экспериментальных данных

Таблица: Результаты проверки нормальности распределений логарифмов отношений времен работы с помощью критерия Шапиро-Уилка

<i>N</i>	ГА / <i>МуАСО</i>
5	0,491
6	0,273
7	0,606
8	0,367
9	0,153
10	0,270

2. Проверка гипотез о средних временах работы алгоритмов

Таблица: Проверка гипотезы: *MuACO* в среднем быстрее, чем ГА

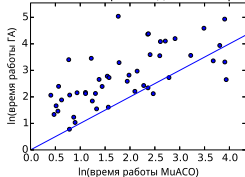
<i>N</i>	<i>p-value</i>
5	$3,47 \cdot 10^{-6}$
6	$1,73 \cdot 10^{-7}$
7	$4,92 \cdot 10^{-5}$
8	$1,73 \cdot 10^{-4}$
9	$1,13 \cdot 10^{-4}$
10	$1,73 \cdot 10^{-4}$

3. Диаграммы логарифмов времен работы

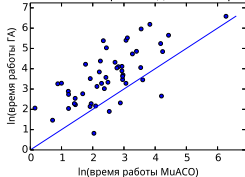
1. MuACO vs ГА
2. SAT+MuACO vs ГА
3. pMuACO, psMuACO, pstMuACO vs параллельный ГА

3.1. MuACO vs ГА

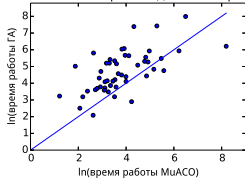
N=5: MuACO быстрее ГА для 86 % примеров



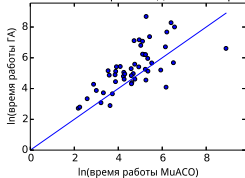
N=6: MuACO быстрее ГА для 90 % примеров



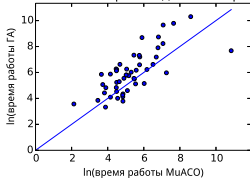
N=7: MuACO быстрее ГА для 84 % примеров



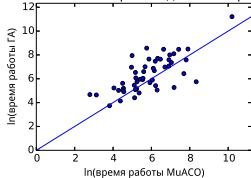
N=8: MuACO быстрее ГА для 76 % примеров



N=9: MuACO быстрее ГА для 76 % примеров

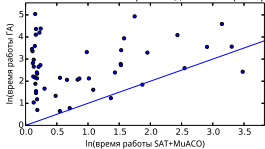


N=10: MuACO быстрее ГА для 72 % примеров

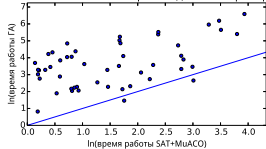


3.2. SAT+MuACO vs ГА

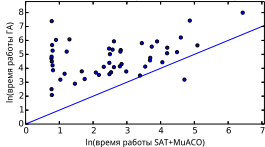
N=5: SAT+MuACO быстрее ГА для 94 % примеров



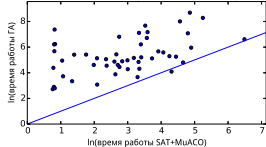
N=6: SAT+MuACO быстрее ГА для 96 % примеров



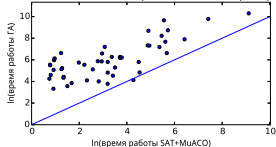
N=7: SAT+MuACO быстрее ГА для 98 % примеров



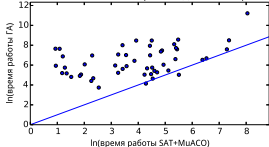
N=8: SAT+MuACO быстрее ГА для 98 % примеров



N=9: SAT+MuACO быстрее ГА для 98 % примеров



N=10: SAT+MuACO быстрее ГА для 96 % примеров



3.3. pMuACO, psMuACO, pstMuACO vs параллельный ГА

