

Эволюционные вычисления и генерация конечных автоматов

Царев Ф. Н., Шалыто А. А.

Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики

Третья российская конференция с международным участием
«Технические и программные средства систем управления,
контроля и измерения»

Пленарный доклад

17 апреля 2012 года

Институт проблем управления имени В. А. Трапезникова РАН
(г. Москва, Россия)

Поисковая инженерия программного обеспечения

- Search-Based Software Engineering (SBSE)
- Область знаний, посвященная применению методов поисковой оптимизации для решения задач программной инженерии
 - *Harman M. Software Engineering Meets Evolutionary Computation // Computer. 2011. Vol. 44, № 11, pp. 31 – 39.*
- Наиболее популярный метод – эволюционные вычисления
 - На июнь 2011 года – 830 публикаций, из которых эволюционные вычисления используются в 68,6%

Публикации по поисковой инженерии ПО

- http://crestweb.cs.ucl.ac.uk/resources/sbse_repository/
- Июнь 2011 года – 830 публикаций
- Апрель 2012 года – 1020 публикаций



<http://www.osp.ru/os/2011/10/13012235/>

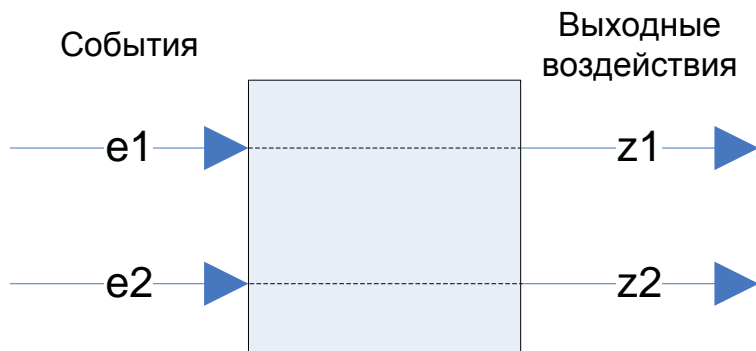
Методы поисковой инженерии ПО

- Эволюционные алгоритмы
- Муравьиные алгоритмы
- Метод роя частиц
- Метод имитации отжига
- Метод спуска
- Алгоритмы оценки распределений
- Поиск с запретами
- Меметические алгоритмы
- Метод рассеянного поиска
- Квадратичное программирование
- Целочисленное программирование
- Искусственные иммунные системы
- ...

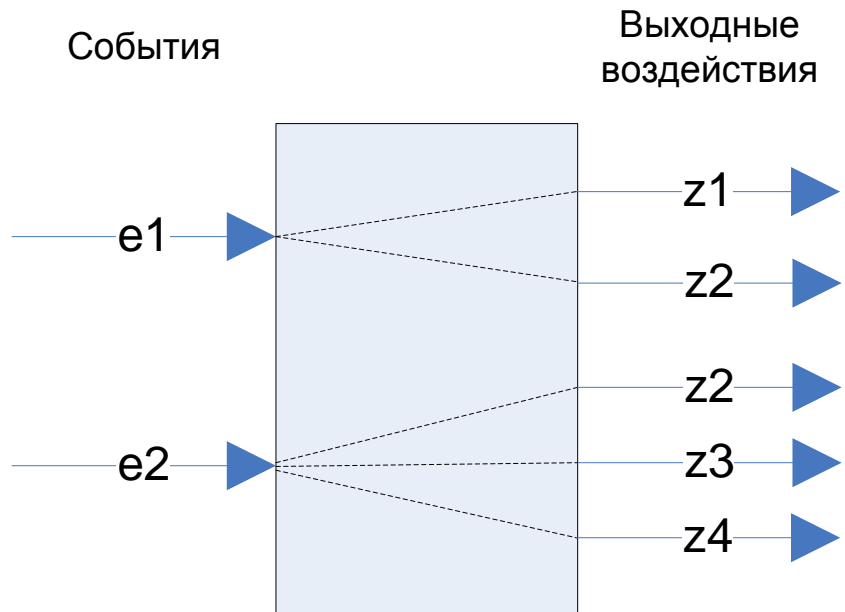
Задачи программной инженерии, решаемые с помощью поисковой оптимизации

- Анализ требований
- Прогнозирование хода разработки
- Проектирование
- Тестирование
- Рефакторинг
- Верификация
- ...

Системы со сложным поведением

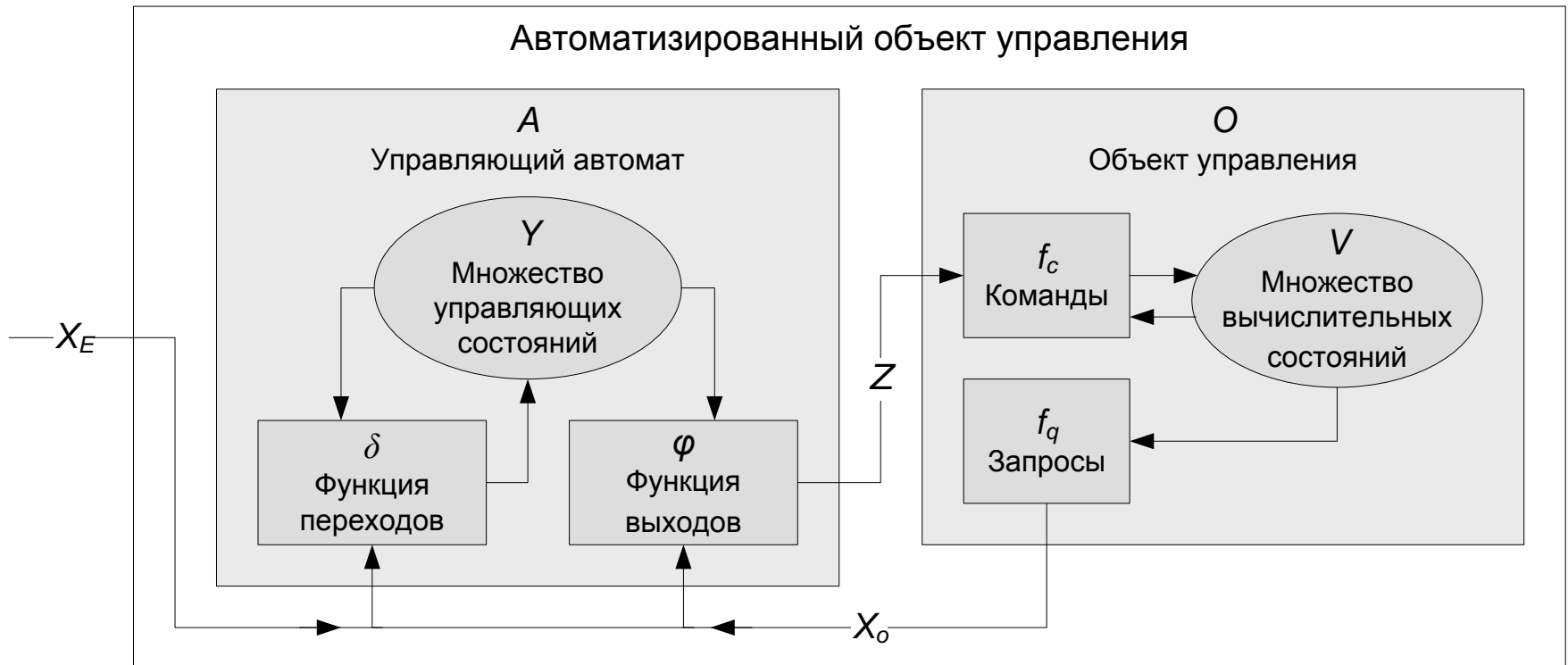


Система с простым поведением



Система со сложным поведением

Автоматное программирование



Достоинства автоматного программирования

- Существенно упрощается верификация программ с использованием метода *Model Checking*
 - Построение модели Крипке по автоматной программе может быть автоматизировано
- При использовании инструментальных средств для поддержки автоматного программирования (например, *UniMod*) более 60% исходного кода программы могут быть сгенерированы автоматически
 - *Шалыто А. А.* Автоматное программирование / Материалы конференции с международным участием «Технические и программные средства систем управления, контроля и измерения» (УКИ'10). ИПУ РАН. 2010, с. 156 – 167.

Решаемая проблема

- Часто эвристическое построение автоматов затруднено
- Построенные вручную автоматы зачастую не оптимальны
- Одно из решений – автоматизированная генерация конечных автоматов с помощью поисковой оптимизации

Генерация автоматов как задача поисковой инженерии ПО

- Относится к задачам проектирования ПО
- Для решения задач проектирования ПО применялись:
 - Эволюционные вычисления
 - Метод имитации отжига
 - Муравьиные алгоритмы

Выбор метода генерации конечных автоматов

- Метод имитации отжига не дает существенного улучшения результатов по сравнению с генетическими алгоритмами
 - *Заикин А. К.* Разработка методов построения конечных автоматов с использованием алгоритма имитации отжига на примере игры «Война за ресурсы» // Научно-технический вестник СПбГУ ИТМО. 2011. № 2, с. 49 – 54.
- Муравьиные алгоритмы более приспособлены для задач, в которых решением является путь в графе
- На основании изложенного были выбраны эволюционные вычисления

Эволюционные алгоритмы для генерации конечных автоматов



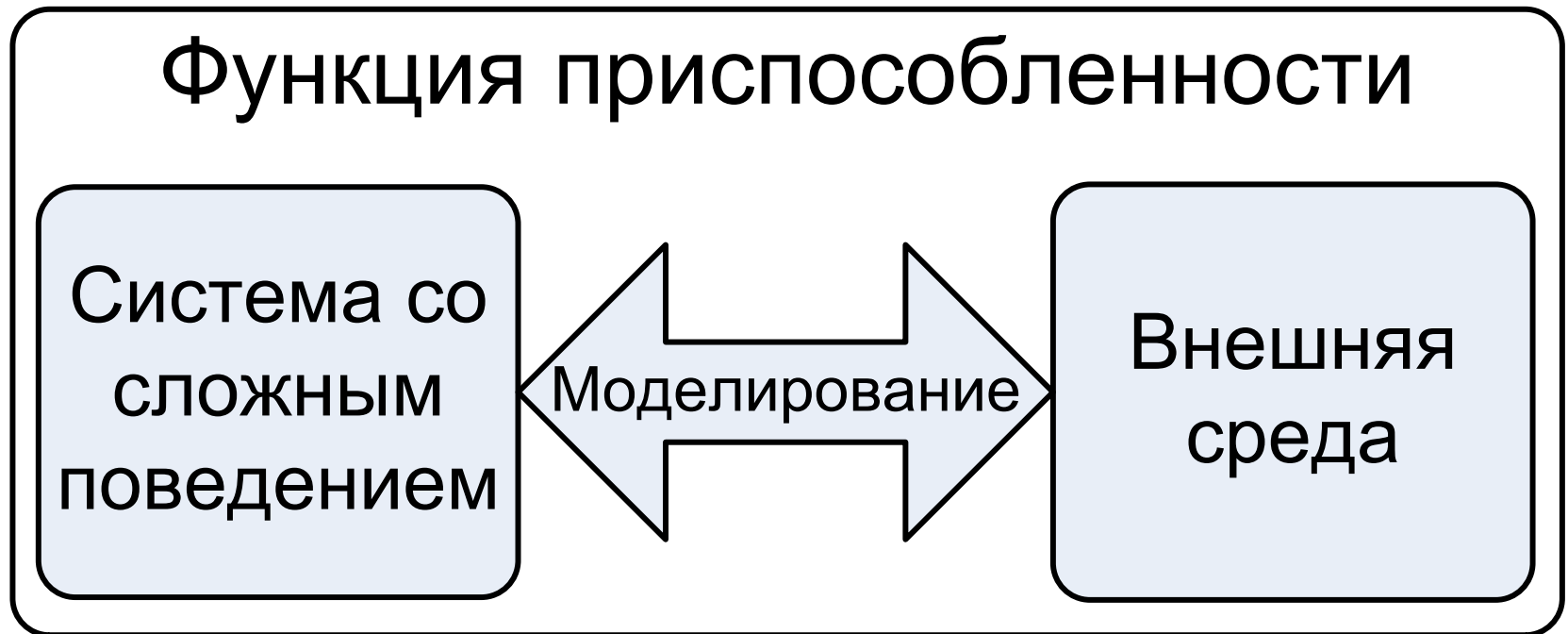
Компоненты эволюционного алгоритма

- Представление конечных автоматов в виде особей
- Операция мутации
- Операция скрещивания
- Для конечных автоматов трудно предложить «осмысленный» способ выполнения операции скрещивания, так как нет четкого понимания того, как в них выделять функциональные блоки
 - *Lucas S., Reynolds J.* Learning Deterministic Finite Automata with a Smart State Labeling Algorithm // IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 27, №7, 2005, pp. 1063–1074.
 - *Johnson C.* Genetic Programming with Fitness based on Model Checking. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007. Volume 4445/2007, pp. 114 – 124.

Методы генерации автоматов

- Методы, использующие моделирование для вычисления функции приспособленности
- Методы, использующие обучающие примеры (тесты) для вычисления функции приспособленности
- Методы, использующие верификацию при вычислении функции приспособленности

Моделирование для вычисления функции приспособленности (1)

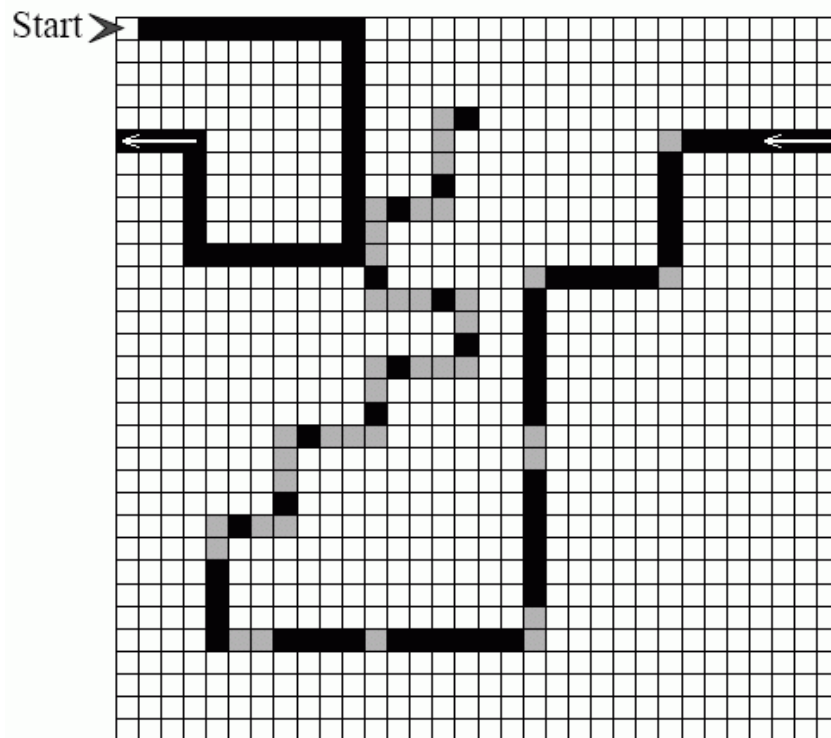


Моделирование для вычисления функции приспособленности (2)

- Первые работы – 60-е годы XX века
 - *Fogel L., Owens A., Walsh M. Artificial Intelligence through Simulated Evolution. NY: Wiley, 1966.*
- Автоматы применялись не в контексте программной инженерии, а в контексте искусственного интеллекта
- Задача построения автомата, предсказывающего битовую последовательность
- «Построение вручную автоматов столь же результативных и простых, как те, что были построены эволюционным алгоритмом, является крайне сложной задачей»

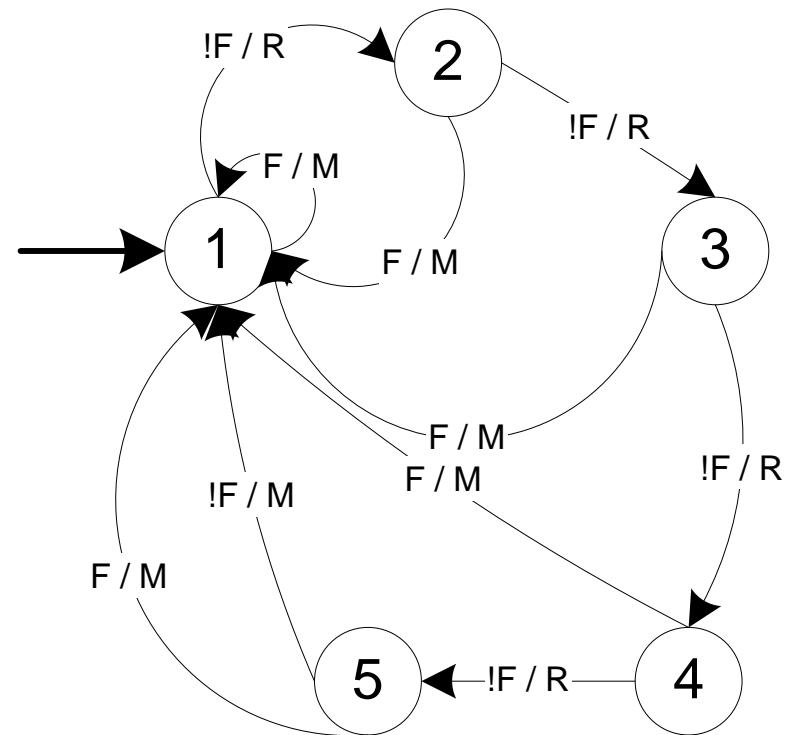
Задача «Умный муравей»

- Тор – 32x32
- 89 клеток с едой
- 200 ходов
- Расположение еды и начальная позиция муравья фиксированы
- Цель – создать муравья, который съест всю еду



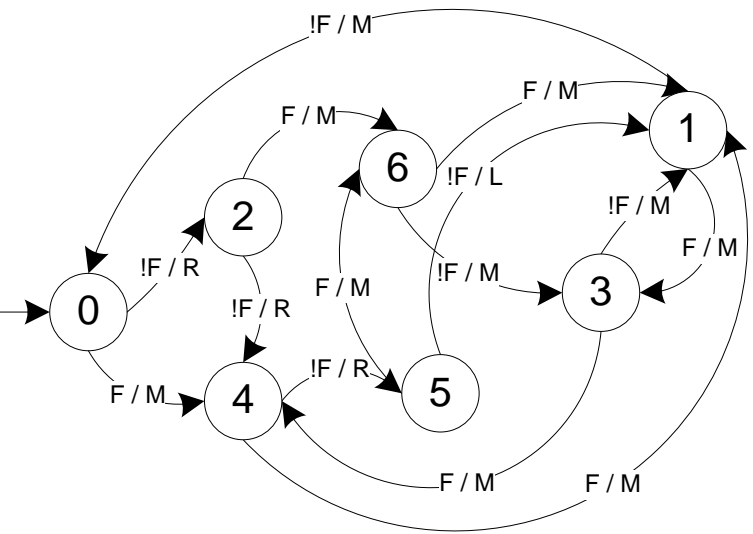
Конечный автомат – способ описания муравья

- Автомат с действиями на переходах (автомат Мили)
- Придумать автомат с большим числом состояний – просто
- С небольшим – сложно
- На рисунке – 5 состояний, за 200 ходов – 81 единица еды



Решение задачи «Умный муравей»

- Известные решения:
 - Автомат из 13 состояний (Jefferson D., Collins R., Cooper C., Dyer M., Flowers M., Korf R., Taylor C., Wang A. The Genesys System. 1992.)
 - Автомат из 11 состояний (Angeline P. J., Pollack J. Evolutionary Module Acquisition // Proceedings of the Second Annual Conference on Evolutionary Programming. 1993.)
 - Автомат из 8 состояний (Chambers L. D. Practical Handbook of Genetic Algorithms, Volume 3, Chapter 26, 6 – Algorithms to Improve the Convergence of a Genetic Algorithm with a Finite State Machine Genome. CRC Press, 1999.)
- Предложен новый подход к выполнению операции скрещивания, учитывающий поведение, задаваемое автоматом
- Построены два автомата из **7 состояний** после перебора 160 и 230 млн. автоматов
- Полный перебор $\sim 3 \cdot 10^{18}$ автоматов



Представление автоматов с большим числом входных переменных

- Деревья решений
 - В каждом состоянии функция переходов и функция действий задается деревом решений
 - *Данилов В. Р.* Метод представления автоматов деревьями решений для использования в генетическом программировании //Научно-технический вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование, с. 103 – 108.
<http://is.ifmo.ru/works/2008/Vestnik/53/08-genetic-automata-decision-tree-method.pdf>
- Сокращенные таблицы переходов
 - В каждом состоянии учитываются значения не всех входных переменных, а только нескольких (как правило, не более пяти) из них
 - *Поликарпова Н. И., Точилин В. Н., Шалыто А. А.* Применение генетического программирования для генерации автоматов с большим числом входных переменных //Известия РАН. Теория и системы управления. 2010. № 2, с.100 – 117.
http://is.ifmo.ru/works/polikarpova_samolet.pdf.

Построение автомата управления моделью самолета

- *Поликарпова Н. И., Точилин В. Н., Шалыто А. А.* Метод сокращенных таблиц для генерации автоматов с большим числом входных переменных на основе генетического программирования // Известия РАН. Теория и системы управления. 2010. № 2, с.100 – 117.
- Для моделирования самолета использовался *open-source* симулятор *FlightGear*
- Вычисление функции приспособленности занимало 5 минут
- Построение конечного автомата занимало несколько недель

Недостатки метода

- Функцию приспособленности необходимо отдельно реализовывать для каждой задачи
- Моделирование, как правило, требует достаточно больших вычислительных ресурсов

Применение обучающих примеров (тестов) для вычисления функции приспособленности

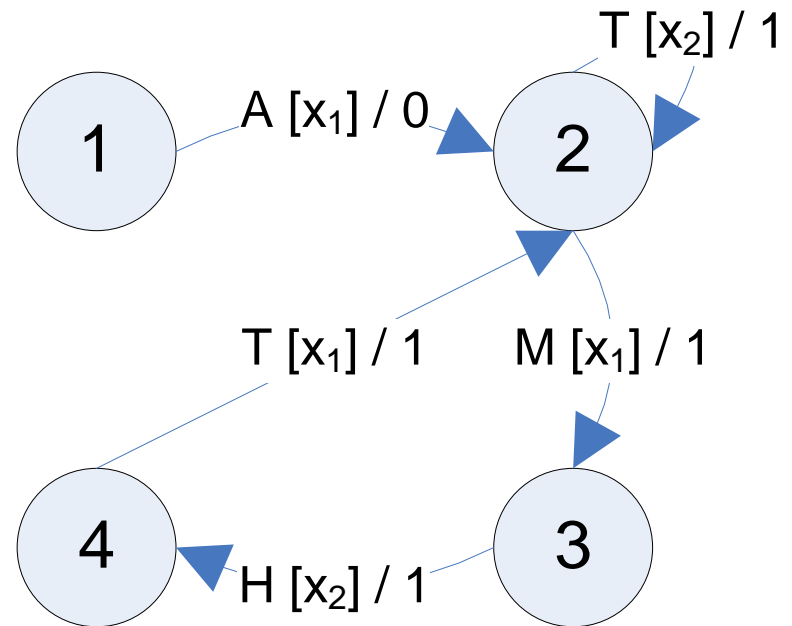


Общее описание метода

- Каждый тестовый пример – пара из входной последовательности событий *Input* и соответствующей ей последовательности выходных воздействий *Answer*
- Функция приспособленности отражает то, насколько «хорошо» автомат работает на тестовых примерах
- Для новой задачи необходимо подготовить только новые тестовые примеры

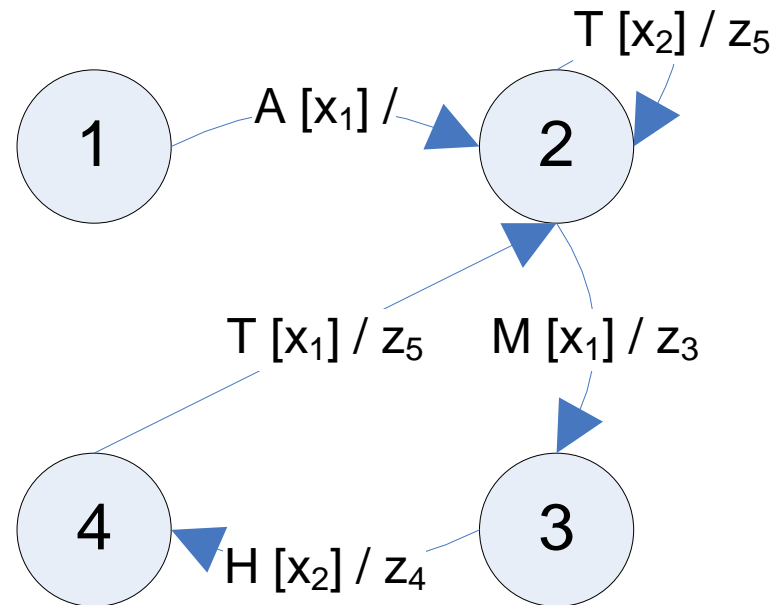
Представление автомата в виде особи эволюционного алгоритма

- Списки переходов для каждого состояния + номер начального состояния
- Для каждого перехода хранится событие, по которому он выполняется, и число выходных воздействий, но не хранятся сами выходные воздействия



Алгоритм расстановки ВЫХОДНЫХ ВОЗДЕЙСТВИЙ

- Каждый переход помечается теми ВЫХОДНЫМИ воздействиями, которые чаще всего встречаются на нем в тестах



Операция скрещивания

- Два варианта – обычное и с учетом тестов
- **Обычное** – для каждого номера состояния проводятся следующие операции:
 - Переходы автоматов-родителей объединяются в общий список
 - Элементы списка переставляются случайным образом
 - Элементы списка распределяются по автоматам-потомкам
- **С учетом тестов** – переходы, которые используются при обработке нескольких тестов, которые автоматы проходят лучше всего, не затрагиваются

Операция мутации

- Каждое из следующих действий производится с вероятностью 5%:
 - Изменение начального состояния
 - Изменение каждого из переходов
 - Изменение на единицу числа переходов для каждого из состояний (уменьшение или увеличение с вероятностью по 0.5)

Функция приспособленности

- Основана на вычислении редакционного расстояния между последовательностью выходных воздействий и «правильным ответом»

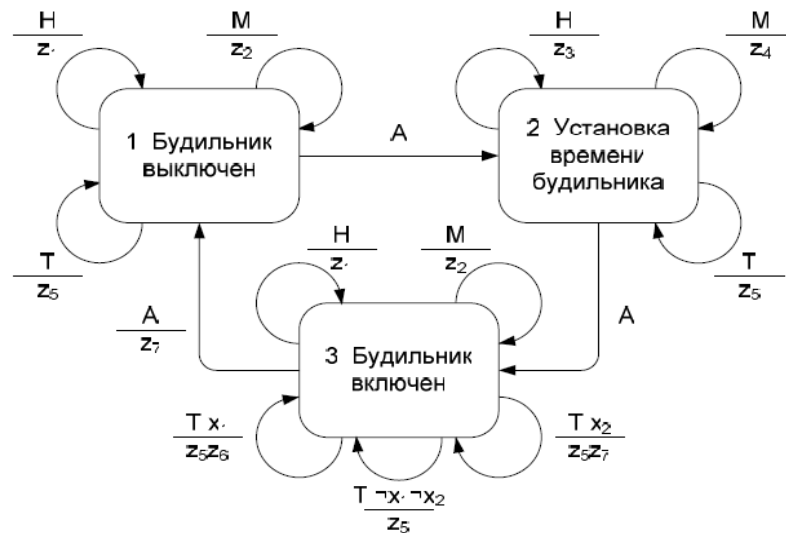
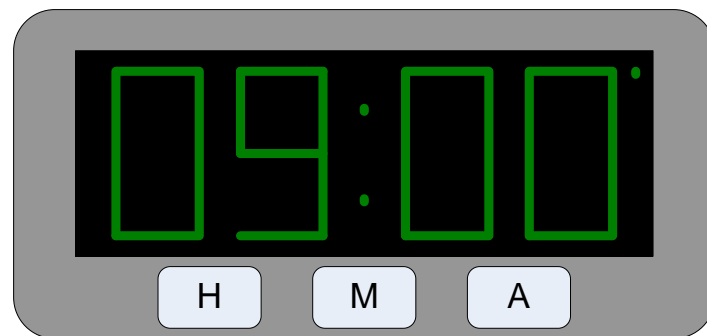
$$FF_1 = \frac{\sum_{i=1}^n \left(1 - \frac{ED(\text{Output}[i], \text{Answer}[i])}{\max(|\text{Output}[i]|, |\text{Answer}[i]|)}\right)}{n}$$

$$FF_2 = \begin{cases} 10 \cdot FF_1 + \frac{1}{M} \cdot (M - cnt), & FF_1 < 1 \\ 20 + \frac{1}{M} \cdot (M - cnt), & FF_1 = 1 \end{cases}$$

ED – редакционное расстояние

Модельная задача – построение автомата управления часами с будильником

- Четыре события
- Две входные переменные
- Семь выходных воздействий
- 38 обучающих примеров
 - Суммарная длина входных последовательностей: 242
 - Суммарная длина выходных последовательностей: 195



Эволюционные алгоритмы

Название	Тип	Традиционное скрещивание	Скрещивание по тестам
МС	МС	0	0
ЭС	ЭС	0	0
ГП-1	ГП	1	0
ГП-2	ГП	0	1
ГП-2+МС	ГП+МС	0	1
ГП-2+ЭС	ГП+ЭС	0	1

Статистика

- Проведено 1000 запусков
- Для каждого запуска записано число вычислений функции приспособленности

Алгоритм	Минимум	Максимум	Среднее	Медиана
МС	1150	9592213	1423982	957746
ЭС	1506	9161811	3447390	856730
ГП-1	855390	38882588	5805943	4588736
ГП-3	32830	599022	117977	83787
ГП+МС-2	26740	188509	53706	48106
ГП+ЭС-2	27656	323935	55603	50599

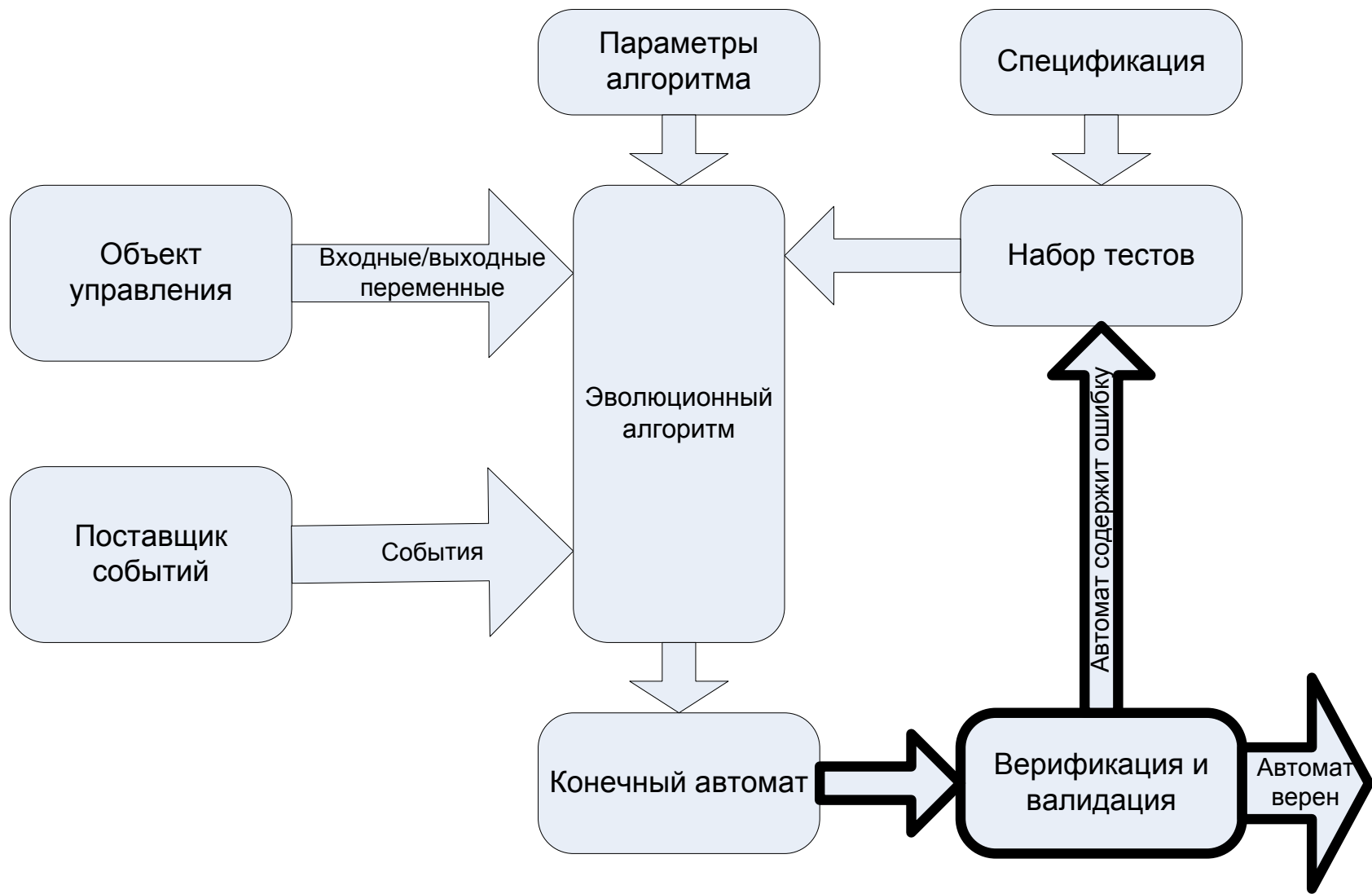
Построение автомата управления моделью самолета (1)

- *Alexandrov A., Sergushichev A., Kazakov S., Tsarev F.* Genetic Algorithm for Induction of Finite Automata with Continuous and Discrete Actions / Proceedings of the 2011 GECCO conference companion on Genetic and evolutionary computation. New York: ACM, 2011, с. 775–778. ISBN: 978-1-4503-0690-4.
- Для моделирования самолета используется симулятор *FlightGear*
- Записаны примеры того, как самолетом управляет человек

Построение автомата управления моделью самолета (2)

- Особь в эволюционном алгоритме – «каркас» автомата
- Функция приспособленности выбрана так, чтобы расстановку выходных воздействий, максимизирующую функцию приспособленности при заданном «каркасе» автомата, можно было найти аналитически – с помощью решения нескольких систем линейных уравнений
- Время вычисления функции приспособленности – 0,2 сек.
- Построен автомат, под управлением которого самолет выполняет «мертвую петлю»

«Тестирование позволяет показать наличие ошибок, но не их отсутствие» (Э. Дейкстра)



Методы, использующие верификацию при вычислении функции приспособленности

- По построению конечных автоматов известна одна работа
 - *Johnson C. Genetic Programming with Fitness based on Model Checking. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007. Volume 4445/2007, pp. 114 – 124.*
 - Для вычисления функции приспособленности применяется только верификация – большой размер пространства поиска
 - Для его сокращения предлагается применять совместно обучающие примеры и верификацию

Методы, использующие верификацию при вычислении функции приспособленности



Функция приспособленности

Учет поведения
автомата на
тестах

$$\longrightarrow FF_1 = \frac{\sum_{i=1}^n \left(1 - \frac{ED(\text{Output}[i], \text{Answer}[i])}{\max(|\text{Output}[i]|, |\text{Answer}[i]|)}\right)}{n}$$

$$FF = FF_1 + FF_1 \cdot \frac{n_1}{n_2} + \frac{1}{M} \cdot (M - \text{cnt})$$

- ED – редакционное расстояние
- n_1 – число верных формул
- n_2 – общее число формул
- M – число, большее максимального числа переходов

Функция приспособленности

Учет поведения
автомата на
тестах

$$\longrightarrow FF_1 = \frac{\sum_{i=1}^n \left(1 - \frac{ED(\text{Output}[i], \text{Answer}[i])}{\max(|\text{Output}[i]|, |\text{Answer}[i]|)}\right)}{n}$$

$$FF = FF_1 + FF_1 \cdot \frac{n_1}{n_2} + \frac{1}{M} \cdot (M - \text{cnt})$$

- ED – редакционное расстояние
- n_1 – число верных формул
- n_2 – общее число формул
- M – число, большее максимального числа переходов

Учет числа переходов

Функция приспособленности

Учет поведения
автомата на
тестах

$$\longrightarrow FF_1 = \frac{\sum_{i=1}^n \left(1 - \frac{ED(\text{Output}[i], \text{Answer}[i])}{\max(|\text{Output}[i]|, |\text{Answer}[i]|)}\right)}{n}$$

$$FF = FF_1 + FF_1 \cdot \frac{n_1}{n_2} + \frac{1}{M} \cdot (M - \text{cnt})$$

- ED – редакционное расстояние
- n_1 – число верных формул
- n_2 – общее число формул
- M – число, большее максимального числа переходов

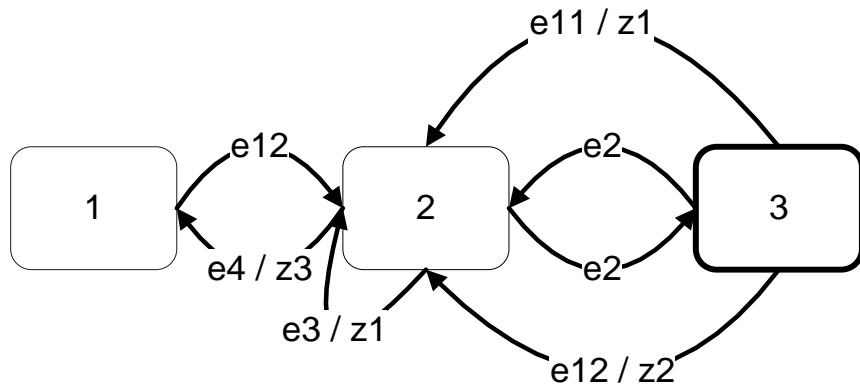
Учет LTL-формул

Пример – система управления дверьми лифта

- Пять событий, три выходных воздействия
- Девять тестов
- Одиннадцать LTL-формул:
 - $\mathbf{G}(\text{wasEvent}(e11) \Rightarrow \text{wasAction}(z1))$ – начать открытие дверей после того, как нажата кнопка «Открыть двери»
 - $\mathbf{G}(\text{wasEvent}(e4) \Leftrightarrow \text{wasAction}(z3))$
 - $\mathbf{G}(\text{wasEvent}(e3) \Rightarrow \text{wasAction}(z1))$
 - $\mathbf{G}(\text{wasEvent}(e11) \Rightarrow \mathbf{X}(\text{wasEvent}(e4) \text{ or } \text{wasEvent}(e2)))$ – если нажали «Кнопку открыть двери», то следующим событием будет либо $e2$ (открывание дверей успешно завершено) или $e4$ (двери заклинило)
 - $\mathbf{G}(\text{wasAction}(z1) \Rightarrow \mathbf{X}(!\text{wasAction}(z1) \mathbf{U}(\text{wasAction}(z2) \text{ or } \text{wasEvent}(e4))))$

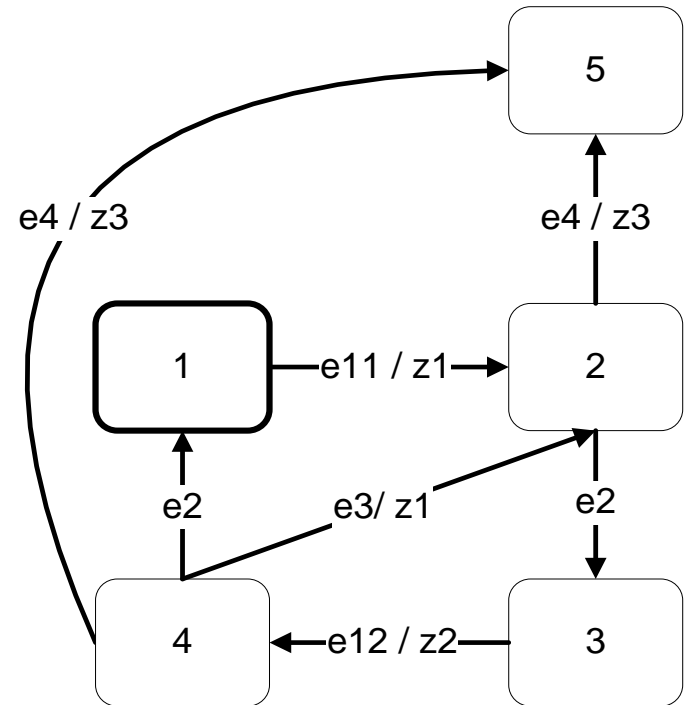
Построенные автоматы

Только тесты



**После поломки
дверей может быть
отдана команда на их
закрытие!**

Тесты + верификация



Результаты экспериментов

- Изменялось число вычислений функции приспособленности
- 1000 экспериментов

	Лифт	
	Тесты	Тесты и формулы
Среднее	7.479×10^4	7.246×10^5
Среднеквадратичное отклонение	2.54×10^4	7.729×10^5
Минимальное	2.184×10^4	7.054×10^4
Максимальное	2.999×10^5	5.492×10^6
Число автоматов, соответствующих спецификации	7	1000

Поддержка исследований

- ФЦП «Научные и научно-педагогические кадры инновационной России на 2009-2013 годы»
 - «Применение методов искусственного интеллекта в разработке управляющих программных систем»
 - «Разработка методов совместного применения генетического и автоматного программирования для построения систем управления беспилотными летательными аппаратами».
 - «Разработка методов машинного обучения на основе генетических алгоритмов для построения управляющих конечных автоматов»
- ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы»
 - «Технология генетического программирования для генерации автоматов управления системами со сложным поведением»
- Грант РФФИ – 10–01–0654-а «Разработка методов машинного обучения на основе генетических алгоритмов для построения управляющих конечных автоматов»

Спасибо за внимание!

tsarev@rain.ifmo.ru

shalyto@mail.ifmo.ru

<http://is.ifmo.ru>