

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

Кафедра компьютерных технологий

**Применение графических процессоров
для генерации управляющих автоматов
на основе моделирования и сценариев работы
с помощью эволюционных алгоритмов**

Автор доклада: Бочкарев А.И.

Научный руководитель: Шалыто А.А.

Обзор предметной области

- Генетические алгоритмы
- Генетическое программирование
- Автоматное программирование
- Автоматическое построение управляющих автоматов

Актуальность

- Скорость построения недостаточно высока
- Долгое вычисление функции приспособленности
- Оптимизации алгоритмов, параллельность
- Ускорение при помощи использования GPU

Цель работы

- Разработать:
 - 1.Модернизированные структуры данных
 - 2.Модернизированные генетические операторы
 - 3.Модернизированные генетические алгоритмы
- Сравнить со старыми реализациями

Сравнение быстродействия

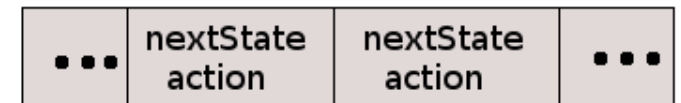
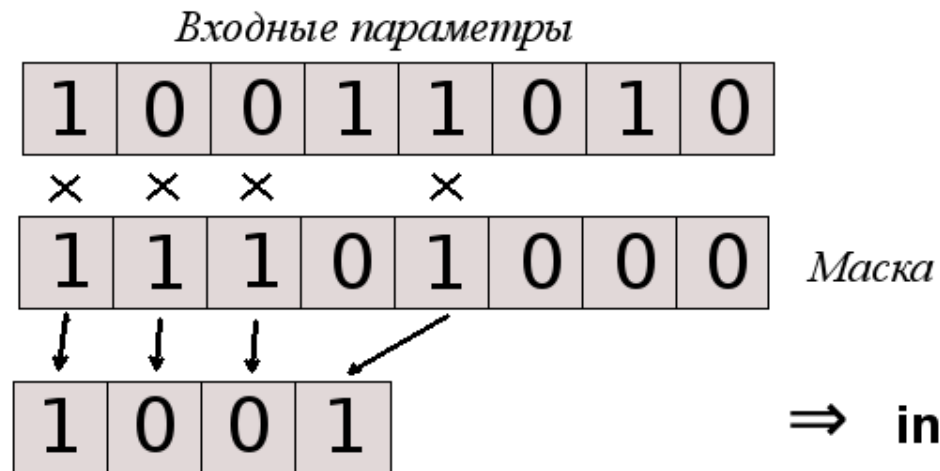
- Оценки эффективности генетических алгоритмов:
 - 1.Время получения результата
 - 2.Число вычислений функции приспособленности
- Оценки быстродействия модернизации:
 - 1.Число поколений, построенных за определенное время
 - 2.Скорость выполнения конкретных операций

Представление автоматов

Полные таблицы переходов



Сокращенные таблицы переходов



⇒ **index=9**

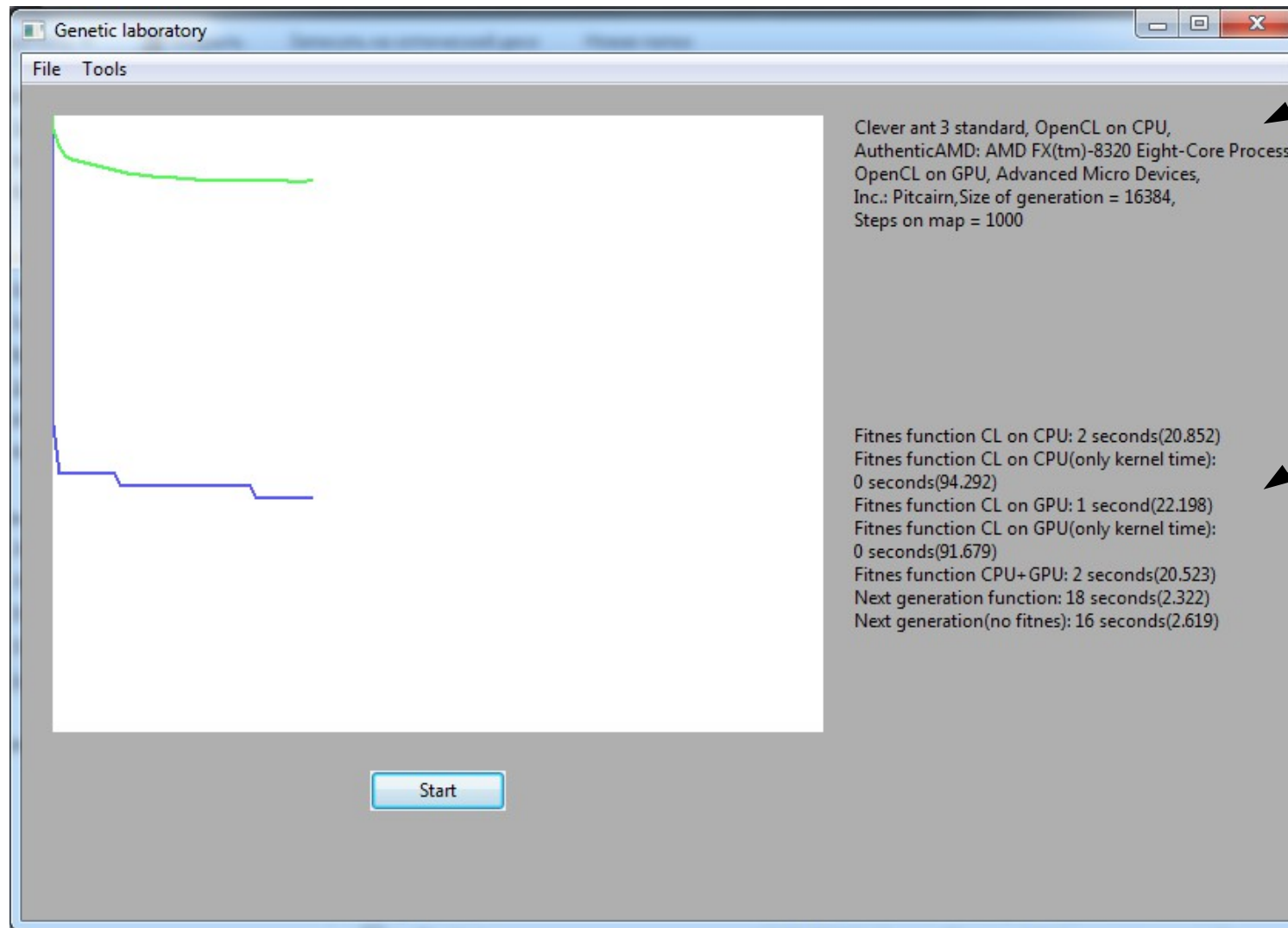
Представление автоматов

Деревья решений

...	N left right	0 next action	0 next action	...
-----	--------------------	---------------------	---------------------	-----

```
position := 0;
for depth := 1 to MAX_DEPTH
{
    currentParam := tree[position].parameter;
    position := currentParam && input[currentParam - 1] ?
tree[position].left : position;
    position := currentParam && !input[currentParam - 1] ?
tree[position].right : position;
}
nextState := tree[position].left;
nextAction := tree[position].right;
```

Виртуальная лаборатория



Информация о платформе

Счетчики производительности

Главное окно лаборатории

Результаты измерений для модификации задачи об умном муравье

- CPU: AMD Phenom II X4 955 3.20 GHz
- GPU: AMD Radeon HD 6850Ti

Размер поколения	Обертка OpenCL	OpenCL на GPU	OpenCL на CPU	C++
1024	7657	12620	7020	1032
3072	2611	5100	2320	337
4098	1960	3400	1760	268
16384	500	560	440	74

Число поколений, построенных за 30 секунд выполнения алгоритма

Результаты измерений для задачи об Умном муравье 3 (сокращенные таблицы)

- CPU: AMD Phenom II X4 955 3.20 GHz
- GPU: AMD Radeon HD 6850Ti

Размер поления	OpenCL на GPU	OpenCL на CPU	C++
3072	264.149	101.7	-
16384	46.890	18.894	-

Скорость вычисления функций приспособленности
(число обработанных поколений в секунду)

Результаты измерений для задачи об Умном муравье 3 (деревья решений)

- CPU: AMD Phenom II X4 955 3.20 GHz
- GPU: AMD Radeon HD 6850Ti

Размер поколения	OpenCL на GPU	OpenCL на CPU	C++
3072	87.625	57.686	-
16384	20.194	10.346	-

Скорость вычисления функций приспособленности
(число обработанных поколений в секунду)

Результаты измерений для задачи об Умном муравье 3 (деревья решений)

- CPU: AMD Phenom II X4 955 3.20 GHz
- GPU: AMD Radeon R9 270

Размер поколения	OpenCL на GPU	OpenCL на CPU	C++	CPU+GPU	
5120	22.4	12.8	3.1	25.2	
16384	5.3	3.8	0.9	7.5	

Скорость вычисления функций приспособленности
(число обработанных поколений в секунду)

Заключение

- Решения, примененные в данной работе, могут быть использованы для решения других задач
- Возможно использовать мощности графического процессора в дополнение к центральному
- Платформа *OpenCL* является мощным инструментом, позволяющим производить эффективные вычисления на различных устройствах

Спасибо за внимание!

Вопросы?