



Язык программирования Scheme

Семинар 6.
10.11.2009



Практическое задание 2

Решение вычислительно сложных задач методами генетического программирования.

Пример задания:

Заданы множество точек на плоскости (P подмножество Z^*Z), положительные числа B и K . Существует ли такое подмножество P_1 множества P , что $|P_1| \leq K$, каждая точка множества $P \setminus P_1$ находится на евклидовом расстоянии, не превосходящем B , от некоторой точки множества P_1 и граф $G = \{P_1, E\}$ связан (в графе G две вершины соединены ребром тогда и только тогда, когда евклидово расстояние между ними не превосходит B)?



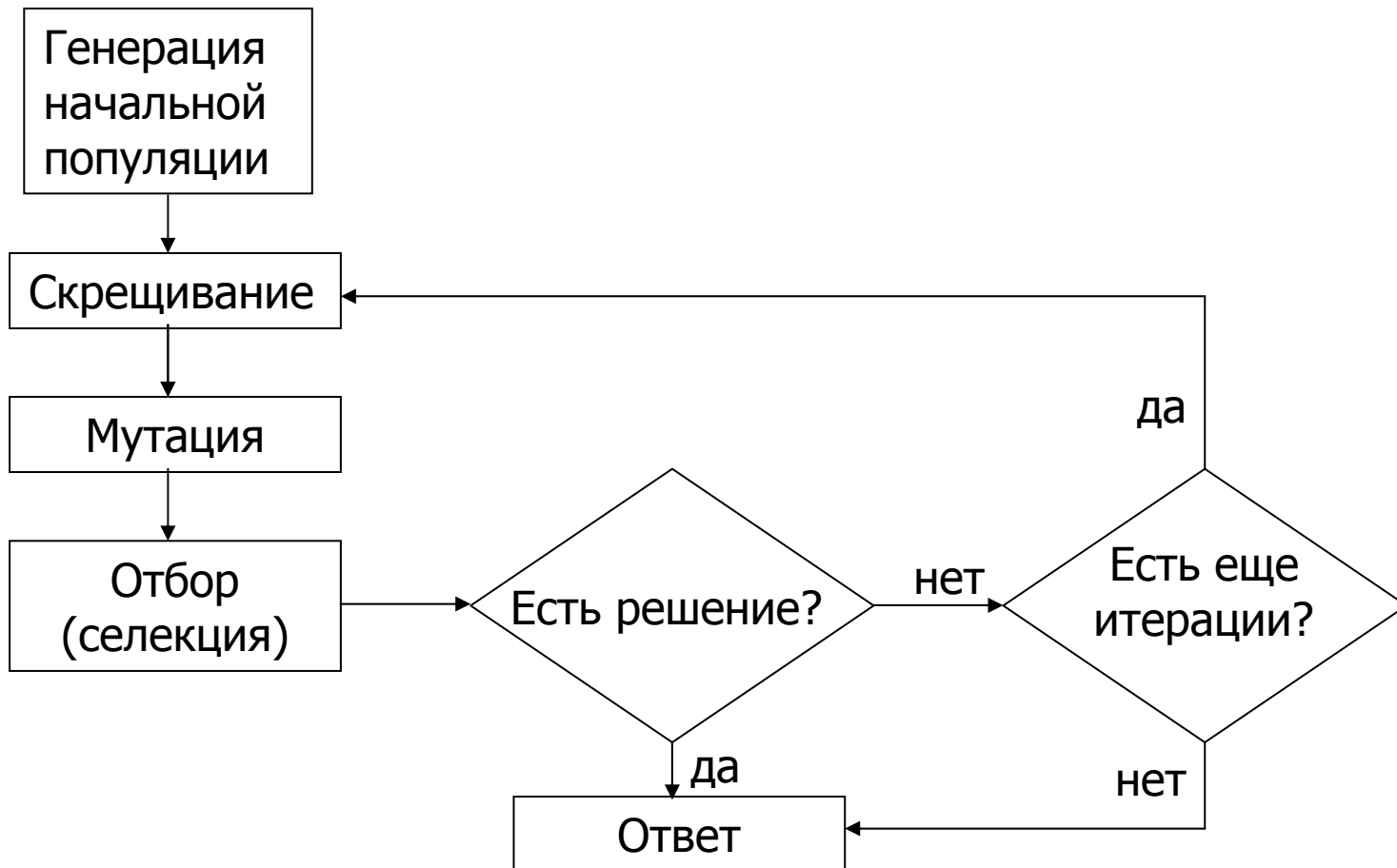
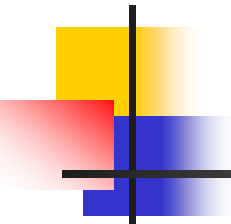
Предварительные проверки

- Если $K=0$, то решения нет
- Если $V=0$, то решения нет
- Если $|P|=0$, решения нет
- Если $|P|=1$, то множества P_1 и P совпадают.
- Если граф $G=\{P,E\}$ не связан, то решения нет



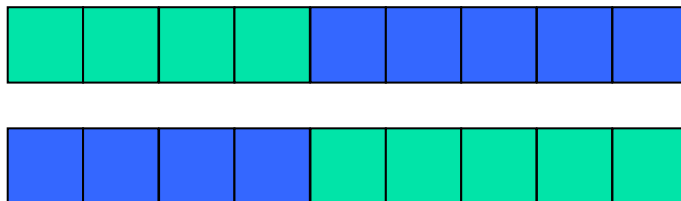
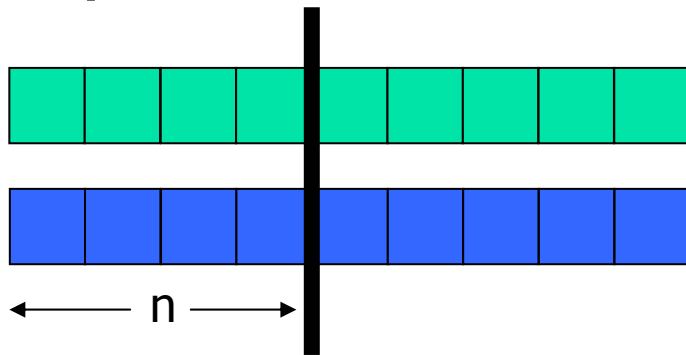
Генетическое программирование

- **Хромосома** – это представление решения задачи
- Множество «кандидатов» решения представляют **популяцию**.
- **Скращивание, мутация** – это преобразование популяции.
- Новая популяция подвергается отбору.
- Все начитается заново!



Скращивание

Выбираются две родительские хромосомы.



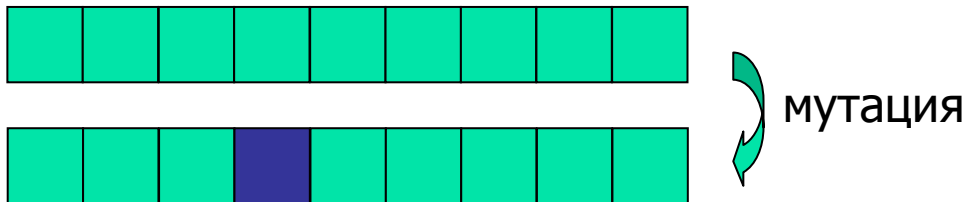
После скрещивания размер популяции увеличивается в 2 раза



Мутация

Номер мутируемого гена выбирается случайным образом

Процент мутируемых хромосом популяции – это **параметр** системы





Оценочная функция

- Оценочная функция позволяет определить степень приспособленности хромосомы.



Отбор (селекция)

- **Метод рулетки.** Отбираются особи посредством n «запусков» рулетки. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер каждого сектора пропорционален значению оценочной функции хромосомы.
- При таком отборе хромосомы с более высокой приспособленностью чаще будут выбираться.



Отбор (селекция)

- Турнирный отбор. Реализуется N турниров. Каждый турнир реализуется для k особей популяции. Среди них выбирается лучшая. Наиболее простая реализация при $k=2$.



Визуализация

- Полученное решение должно быть отрисовано. Рекомендации по визуализации можно посмотреть на сайте unicorn.ctmc.msu.ru/5sem



Математическое моделирование

- Требуется промоделировать зависимость количества итераций от
 - Вероятности мутации
 - Метода отбора
 - Размера начальной популяции (зависит от варианта)
- Выводы!!!



Отчет

- Постановка задачи
- Описание хромосомы
- Описание реализации скрещивания
- Описание реализации мутации
- Описание реализации отбора
- Описание оценочной функции
- Описание визуализации
- Математическое моделирование
- Руководство пользователя



Пример

- Хромосома – список из 0 и 1 длиной $|P|$. 0 – вершина не включается во множество P_1 , 1 – включается. Множество P_1 состоит из вершин, отмеченных 1.



Пример

- Оценочная функция:

Для каждой хромосомы вычисляется $|P1|$;

- Если $|P1| \geq K$, то значение оценочной функции равно разности $|P|$ и $|P \setminus P1|$, для которых в множестве $P1$ есть точка, расстояние до которой не превышает евклидово
- Если $|P1| < K$, то значение оценочной функции равно разности $|P|$ и $|P \setminus P1|$, для которых в множестве $P1$ есть точка, расстояние до которой не превышает евклидово, плюс разность $|P|$ и K
- Если оценочная функция равна 0, то хромосома является решением
- Если граф $G = \{P1, E\}$, то оценочная функция принимает максимально возможное значение $|P|$.



Использование лямбда исчисления на практике.

- Очищенные типы данных в задаче информационной безопасности (Refinement Types) **Andrew Gordon.**
 - Выполнение формального доказательства корректности работы протоколов обмена критических данных.

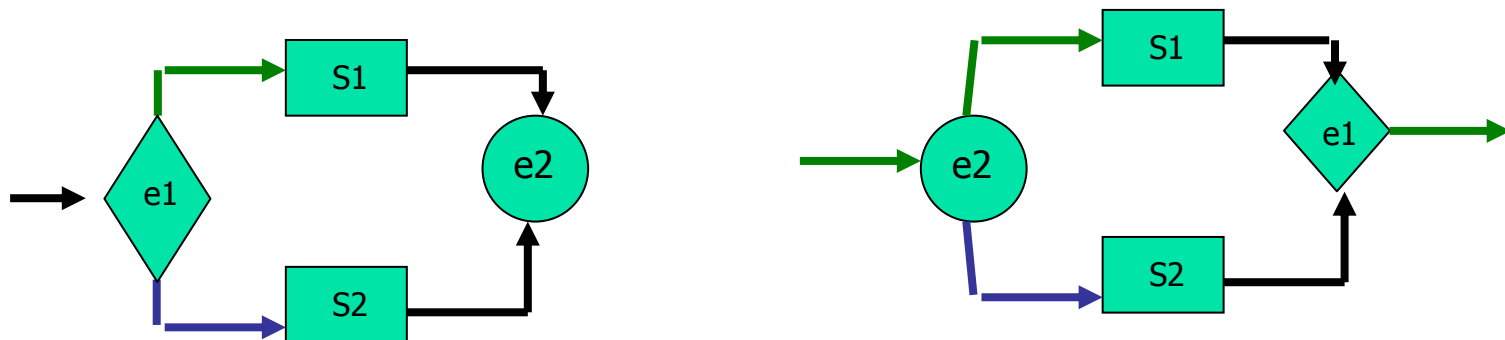


Функциональные языки на практике.

- Доказательство программ и верификация. **Xavier Leroy**.
 - Автоматическое доказательства свойств программ. Инструмент Coq.
 - Исходная программа транслируется в IMP=> автоматическая проверка свойств программы
 - Coq+документация <http://coq.intra.fr/>

Функциональные языки на практике.

- Обратимые программы (Reverting programs) **Robert Gluck**.
 - Программный язык Janus.
 - Условие и цикл



- e1 – test, e2 - assertion

Пример обратимой программы

procedure fib

If $n=0$ then $x1+=1$

$x2+=1$

else $n-=1$

call fib

$x1+=x2$

$x1<=>x2$

fi $x1=x2$

procedure main_fwd

$n+=4$

call fib

procedure main_bwd

$x1+=5$

$x2+=8$

uncall fib