

Эволюционный алгоритм с оптимальной рекомбинацией для задачи аппроксимации булевых функций

Захаров А. О.

Институт математики им. С.Л. Соболева СО РАН (Омский филиал)

Семинар "Модели и алгоритмы для задач составления расписаний"

Работа выполнена при финансовой поддержке РФФ,
проект № 22-71-10015.

Задачи оптимизации с древовидным представлением решений

- задачи построения нелинейных моделей (математических выражений, функций, алгоритмов, программ) на основе заданных экспериментальных данных, множества переменных, базовых функций и операций
- синтез решающих деревьев
- идентификация паттернов в семействах белков и других биопоследовательностей

Генетическое программирование

В алгоритме генетического программирования популяция деревьев итеративно преобразуется посредством операторов воспроизведения, аналогичных процессам селекции, кроссинговера (рекомбинации), мутации и локальных улучшений в живой природе и социумах^а.

^аKoza J.R., Poli R.: Genetic programming (2005)

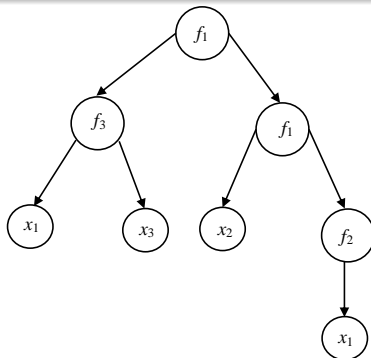
- 1 Koza J.R., Poli R. (2005) Genetic Programming. In: Burke E.K., Kendall G. (eds) Search Methodologies. Springer, Boston, MA.
- 2 Poli, R., Page, J. Solving High-Order Boolean Parity Problems with Smooth Uniform Crossover, Sub-Machine Code GP and Demes. Genetic Programming and Evolvable Machines 1, 37-56 (2000).
- 3 Langdon, W.B. Size Fair and Homologous Tree Crossovers for Tree Genetic Programming. Genetic Programming and Evolvable Machines 1, 95-119 (2000).
- 4 Poli, R., Page, J. Solving High-Order Boolean Parity Problems with Smooth Uniform Crossover, Sub-Machine Code GP and Demes. Genetic Programming and Evolvable Machines 1, 37-56 (2000).
- 5 Moraglio A., Krawiec K., Johnson C.G. (2012) Geometric Semantic Genetic Programming. PPSN 2012. Lecture Notes in Computer Science, vol 7491. Springer, Berlin, Heidelberg.
- 6 Santoso L.W., Singh B., Rajest S.S., Regin R., Kadhim K.H. A Genetic Programming Approach to Binary Classification Problem. EAI Endorsed Transactions on Energy Web. V. 8 (31) (2020).

Представление решений

Функциональное дерево $T = (V, E)$,

Листья из множества $X = \{x_1, x_2, \dots, x_m\}$,

Вершины из множества базовых функций $\mathcal{F} = \{f_1, f_2, \dots, f_k\}$



$X = \{x_1, x_2, x_3\}$,

$\mathcal{F} = \{f_1, f_2, f_3\}$

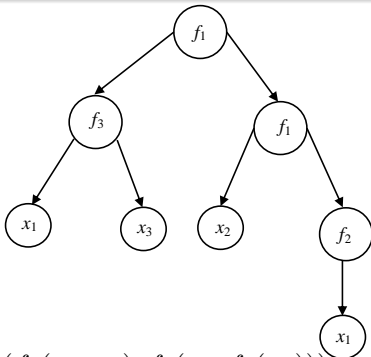
Задача оптимизации

Вход: набор пар $\{(\bar{x}^i, y^i)\}$,

$\bar{x}^i = (\bar{x}_1^i, \dots, \bar{x}_m^i)$, $i = 1, \dots, n$. n - объем обучающей выборки

Целевая функция $g(T) = \sum_{i=1}^n (y_i - T(\bar{x}_m^i))^2$,

$T(\bar{x}^i)$ - значение функционала на дереве T в \bar{x}^i



$$T(x_1, x_2, x_3) = f_1(f_3(x_1, x_3), f_1(x_2, f_2(x_1)))$$

Эволюционные алгоритмы (генетическое программирование)

Основной принцип работы ЭА основан на компьютерном моделировании процесса эволюции с учетом факторов изменчивости, наследования и отбора наиболее приспособленных особей.

Стационарная схема. Схема алгоритма

1. Построить начальную популяцию.
2. Повторять шаги 2.1-2.4 пока не выполнен критерий остановки:
 - 2.1 Выбрать двух родителей $\mathbf{T}_1, \mathbf{T}_2$ из текущей популяции.
 - 2.2 Применить мутацию к \mathbf{T}_1 и \mathbf{T}_2 с вероятностью p_m .Получить \mathbf{T}'_1 и \mathbf{T}'_2
 - 2.3 Построить потомка \mathbf{T}' с помощью скрещивания \mathbf{T}'_1 и \mathbf{T}'_2 .
 - 2.4 Заменить \mathbf{T}' одну из «худших» особей в популяции.
3. Вернуть в качестве результата лучшее по целевой функции дерево за все время работы алгоритма.

Эволюционные алгоритмы (генетическое программирование)

Популяционная схема с элитой. Схема алгоритма

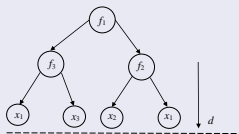
1. Построить начальную популяцию.
2. Повторять шаги 2.1-2.2 пока не выполнен критерий остановки:
 - 2.1 Построить новую подпопуляцию P' , положив первоначально $P' = \{\}$. Повторить шаги 2.1.1-2.1.3 k раз:
 - 2.1.1 Выбрать двух родителей T_1, T_2 из текущей популяции.
 - 2.1.2 Применить мутацию к T_1 и T_2 с вероятностью p_m . Получить T'_1 и T'_2
 - 2.1.3 Построить потомка T' с помощью скрещивания T'_1 и T'_2 . Добавить T' к P' .
 - 2.2 Заменить особями из P' k «худших» особей в популяции.
3. Вернуть в качестве результата лучшее по целевой функции дерево за все время работы алгоритма.



Генерация начальной популяции

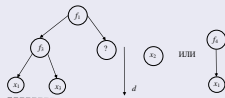
Метод Full

Полное дерево заданной глубины



Метод Crow

В каждой вершине: поддерево или лист с заданной вероятностью
Верхняя граница на глубину

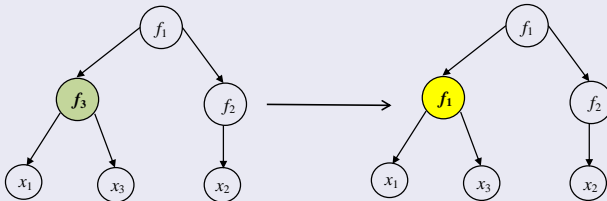


Модификация метода Crow

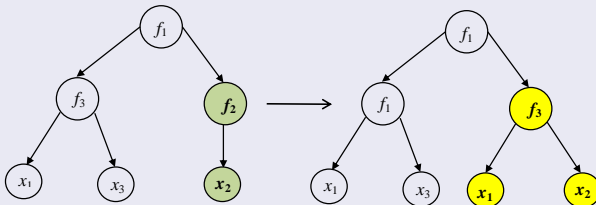
В каждой вершине: поддерево или лист с заданной вероятностью
Верхняя граница на глубину, макс. число вершин
Нижняя граница на глубину

Операторы мутации для деревьев

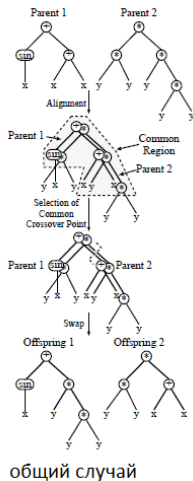
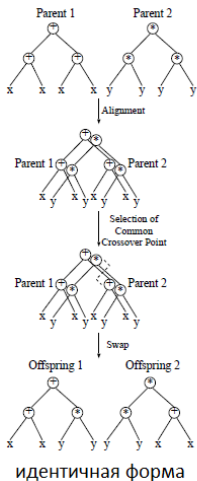
Точечная мутация GP-PM



Замена поддерева GP-SM

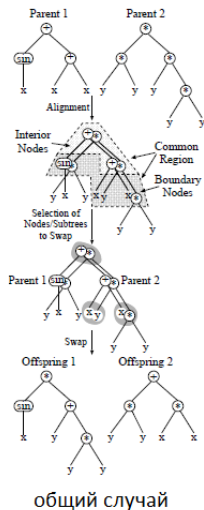
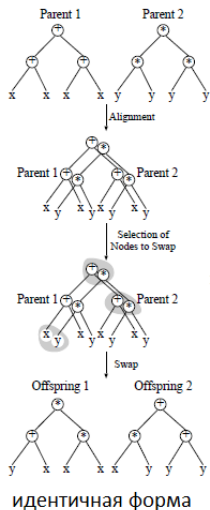


Одноточечный кроссинговер на деревьях GP-SPX



Poli R., Langdon W.B. On the search properties of different crossover operators in genetic programming (1998)

Равномерный кроссинговер на деревьях GP-UX



Poli R., Page J. Solving high-order Boolean parity problems with smooth uniform crossover, sub-machine code GP and demes (2000)

Равномерный кроссинговер на деревьях GP-UX

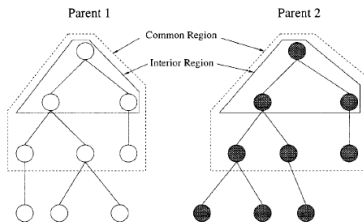


Figure 1. Two parental parse trees prior to GP-UX.

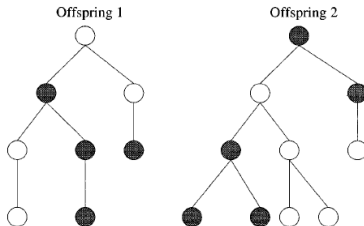


Figure 2. Offspring trees after GP-UX.

Poli R., Page J. Solving high-order Boolean parity problems with smooth uniform crossover, sub-machine code GP and demes (2000)

Задача оптимальной рекомбинации (ЗОР)

Определение основано на свойстве передачи генов.¹

Задача оптимальной рекомбинации

^a**Дано:** индивидуальная задача комбинаторной оптимизации I с множеством допустимых решений Sol и два родительских решения

$\mathbf{p}^1 = (p_1^1, \dots, p_l^1), \mathbf{p}^2 = (p_1^2, \dots, p_l^2)$ из Sol .

Найти: допустимое решение (потомка) $\mathbf{p}' \in Sol$ такое что

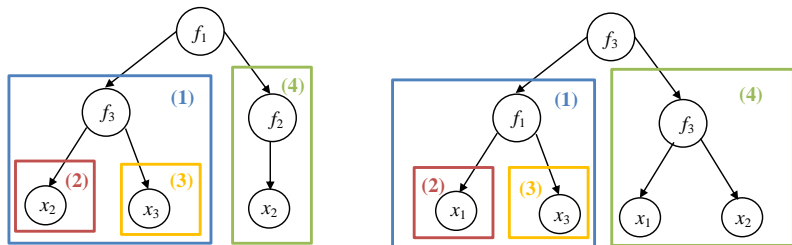
- 1 $p'_j = p_j^1$ или $p'_j = p_j^2 \quad \forall j = 1, \dots, l,$
- 2 для каждого $\bar{\mathbf{p}} \in Sol$ такого что $\bar{p}_j = p_j^1$ или $\bar{p}_j = p_j^2 \quad \forall j$ имеет место

$$f(\mathbf{p}') \leq f(\bar{\mathbf{p}})$$

(в случае задачи на минимум).

^aA.V. Eremeev, J.V. Kovalenko. Optimal recombination in genetic algorithms for combinatorial optimization problems (2014)

¹Radcliffe, N.J.: The algebra of genetic algorithms (1994)



Рассматриваемые виды скрещивания:

Одноточечный кроссинговер: 4 варианта - (1), (2), (3), (4).

Строгий кроссинговер: 3 варианта - (2), (3), (4).

Равномерный (строгий) кроссинговер, 2^3 вариантов - комбинации из (2), (3), (4).

Эксперимент на булевых деревьях

Деревья $T = (V, E)$

Листья $X = \{x_1, x_2, \dots, x_m\}$, $x_i \in \{0, 1\}$, $i = 1, 2, \dots, m$.

Базовые функции $\mathcal{F} = \{\wedge, \vee, \neg\wedge, \neg\vee\}$.

Тестовые данные

1. Even-3 parity
2. Мультиплексор (3-MUX)
3. Случайно сгенерированные значения

even-parity:	0	0	0	0	3-MUX:	0	0	0	0
	0	0	1	1		0	0	1	0
	0	1	0	1		0	1	0	1
	0	1	1	0		0	1	1	1
	1	0	0	1		1	0	0	0
	1	0	1	0		1	0	1	1
	1	1	0	0		1	1	0	0
	1	1	1	1		1	1	1	1

Начальная популяция: метод Full глубины d

Турнирная селекция

Мутация: замена поддерева (GP-SM)

LS: first improvement с неполным просмотром окрестности (GP-SM)

Кроссинговер:

OPT_STR_SINGLE_POINT, OPT_SINGLE_POINT,
RAND_STR_SINGLE_POINT, RAND_SINGLE_POINT,
OPT_STR_UNIFORM

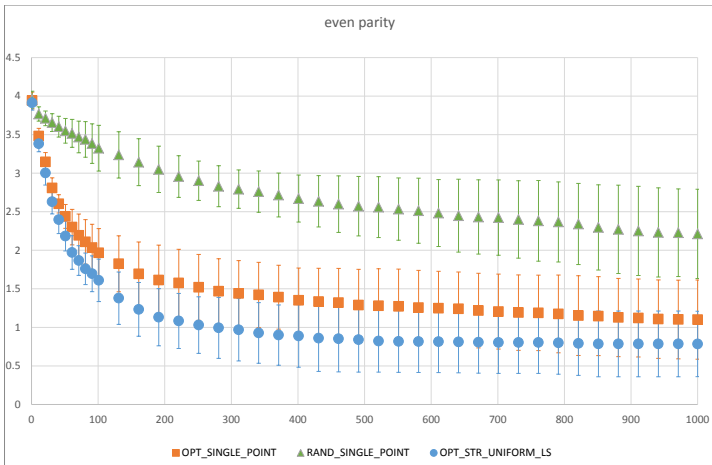
Схемы экспериментов

1. SUBTREE + OPT_STR_SINGLE_POINT
2. SUBTREE + OPT_SINGLE_POINT
3. SUBTREE + RAND_STR_SINGLE_POINT
4. SUBTREE + RAND_SINGLE_POINT
5. LS (SUBTREE) + OPT_STR_UNIFORM

Результаты эксперимента, even-3 parity

1000 итераций, 30 запусков

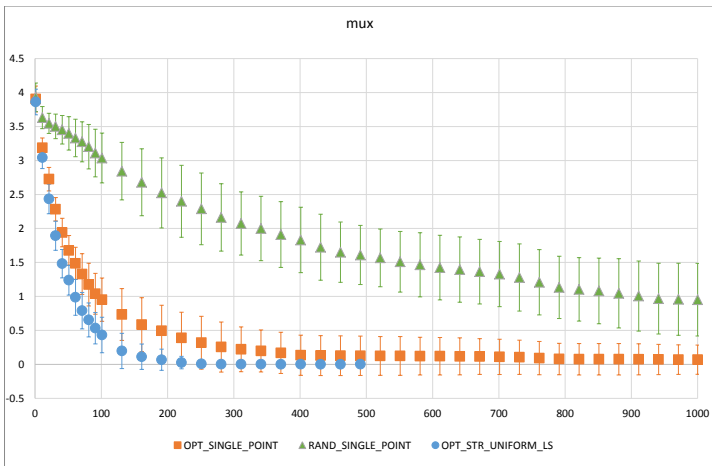
График: среднее значение целевой функции на популяции



Результаты эксперимента, 3-мех

1000 итераций, 30 запусков

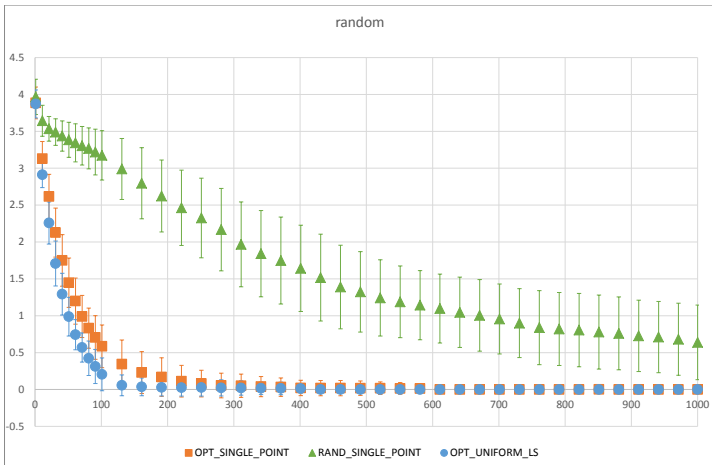
График: среднее значение целевой функции на популяции



Результаты эксперимента, случайным образом

1000 итераций, 30 запусков

График: среднее значение целевой функции на популяции



Доля запусков, в которых получено оптимальное значение целевой функции за равное время работы алгоритма.

	OPT_STR_SINGL	OPT_SINGL	RAND_STR_SINGL	RAND_SINGL	OPT_UNI
even-3 parity	0 (0.87*)	0.4 (1*)	0 (0.43*)	0 (0.5*)	0.53 (1*)
3-mux	0.93	1	0.57	0.77	1
rand	0.93	1	0.7	0.83	1

* доля запусков, когда получено значение, отличное на 1 от оптимального

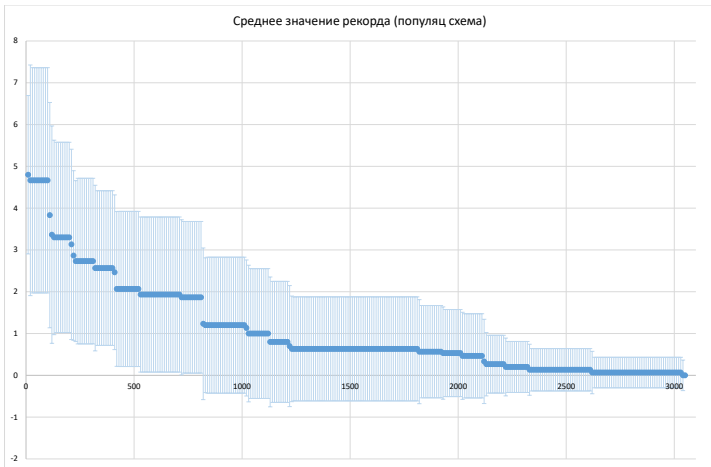
Популяционная схема: элита 70% от популяции

Перезапуск: нет изменений рекорда за p_{size} итераций

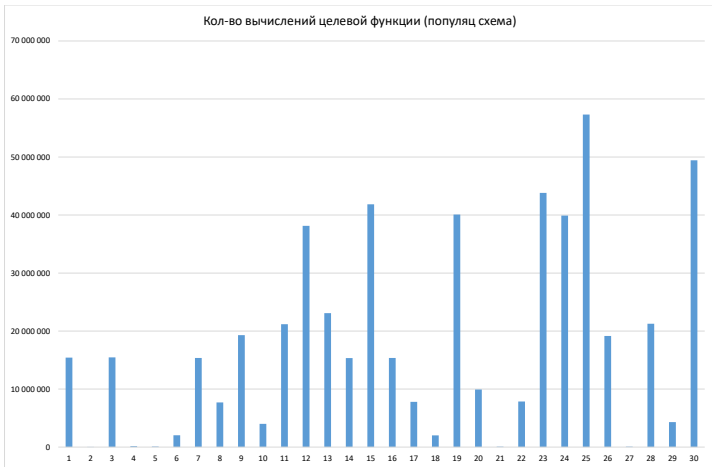
Останов: найден оптимум или достигнуто макс. число итераций

задача	оптимум	верх. оценка	схема	кол-во итераций	размер популяции	перезапуск	кол-во шагов ЛП	макс пар в ОР	кол-во оптимумов	ср. знач. рекорда
6-мих	0	64 стац		1000	30 -		5	15	0	9,43
6-мих	0	64 стац		1000	50 -		5	15	1	7,86
6-мих	0	64 стац		1000	100 -		5	15	1	8,56
6-мих	0	64 стац		1000	30 -		10	7	0	7,96
6-мих	0	64 стац		1000	50 -		10	7	1	5,2
6-мих	0	64 стац		1000	100 -		10	7	0	5,9
6-мих	0	64 стац		3000	100 -		10	7	8	2,6
6-мих	0	64 поп		10000 (3040)	100 +		10	7	30	0
6-мих	0	64 стац		70000	100 +		10	7	27	0,2

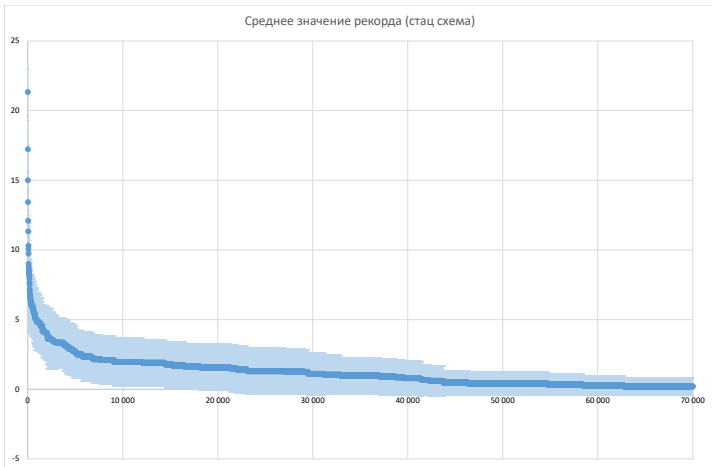
Эксперимент с 6-тих



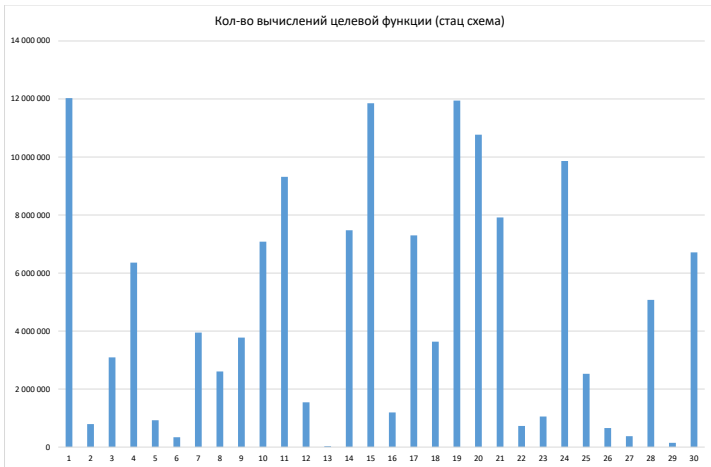
Эксперимент с 6-тих



Эксперимент с 6-тих



Эксперимент с 6-тих



- 1 Рассмотрены задачи оптимизации с представлением решений в виде деревьев. Алгоритм генетического программирования.
- 2 Исследована задача оптимальной рекомбинации (ЗОР) на деревьях.
- 3 Проведен вычислительный эксперимент на тестовых задачах с булевыми деревьями и функциями: even-parity, mix, rand. Проанализирована работа рандомизированных и оптимизированных операторов скрещивания.

Спасибо за внимание!

<https://gitlab.com/alex2108/tree-crossover>