Алгоритмы оценки распределений и байесовской оптимизации

Семинар по вычислительному интеллекту

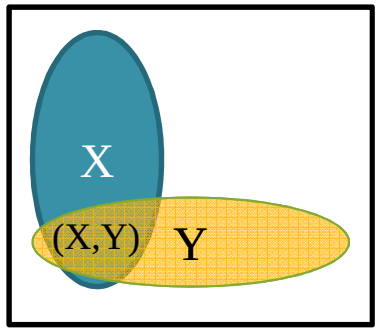
Докладчик: Юрий Цой

г. Томск, 29 октября, 2011 г.

http://qai.narod.ru/TomskWorkshop/index.html

Предварительные замечания

- Совместное распределение и условная вероятность



Независимые СВ $p(X,Y) = p(X)p(Y)$

Условная вероятность $p(X \mid Y) = \dfrac{p(X,Y)}{p(X)}$

Зависимые СВ $p(X,Y) = p(X \mid Y)p(Y) = p(Y \mid X)p(X)$

Многомерный общий случай:

$$p(X_1, X_2, \ldots, X_n) = p(X_1 \mid X_2, X_3, \ldots, X_n)p(X_2 \mid X_3, \ldots, X_n)\ldots p(X_{n-1} \mid X_n)p(X_n)$$

Закон Байеса $p(Y \mid X) = \dfrac{p(X \mid Y)p(Y)}{p(X)}$

Предварительные замечания

- Метод максимального правдоподобия

$$\hat{p}(X_1, X_2, ..., X_n) = \hat{p}(X_1, X_2, ..., X_n \mid \theta) \to \max$$

$$\hat{p}(X_1 X_2, X_3, ..., X_n \mid \theta) = \prod_{i=1}^{n} p(X_1 \mid \Pi_1, \theta)$$

$$\frac{d}{d\theta} \hat{p}(X_1 X_2, X_3, ..., X_n \mid \theta) = 0$$

- Эквивалентная задача

$$\ln \hat{p}(X_1 X_2, X_3, ..., X_n \mid \theta) = \sum_{i=1}^{n} \ln p(X_1 \mid \Pi_1, \theta) \to \max$$

$$\frac{d}{d\theta} \ln \hat{p}(X_1 X_2, X_3, ..., X_n \mid \theta) = 0$$

Введение

Estimation of Distribution Algorithms (EDA)

Probabilistic Model-Building Genetic Algorithms (PMBGA)

Iterated Density Estimation Algorithms (IDEA)

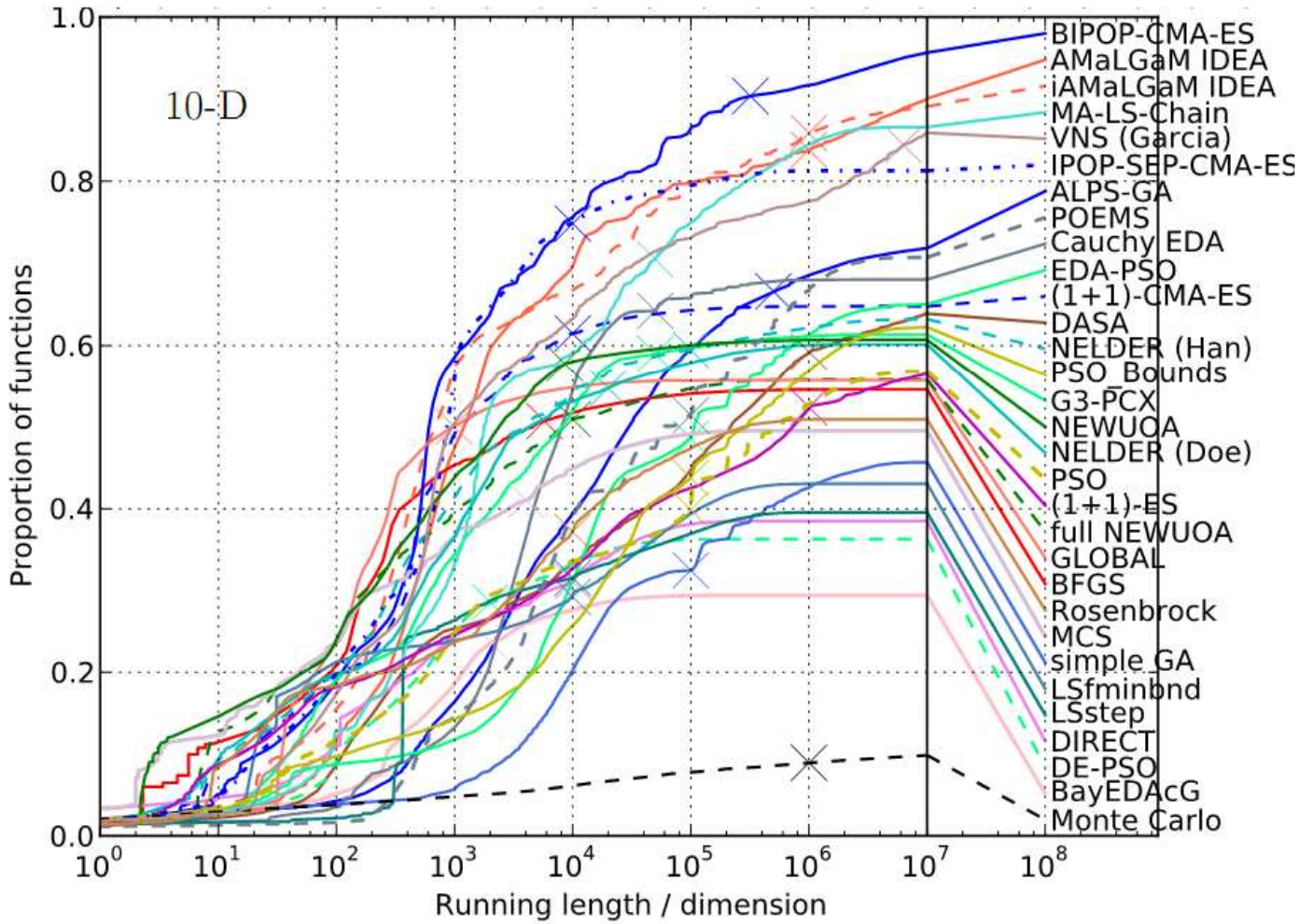
Алгоритмы оценки распределений (АОР) = Эволюционные вычисления + Машинное обучение.

Важный и логичный этап развития эволюционных вычислений, машинного обучения и стохастической оптимизации.

Хороший обзор: [Hauschild, Pelikan, 2011].

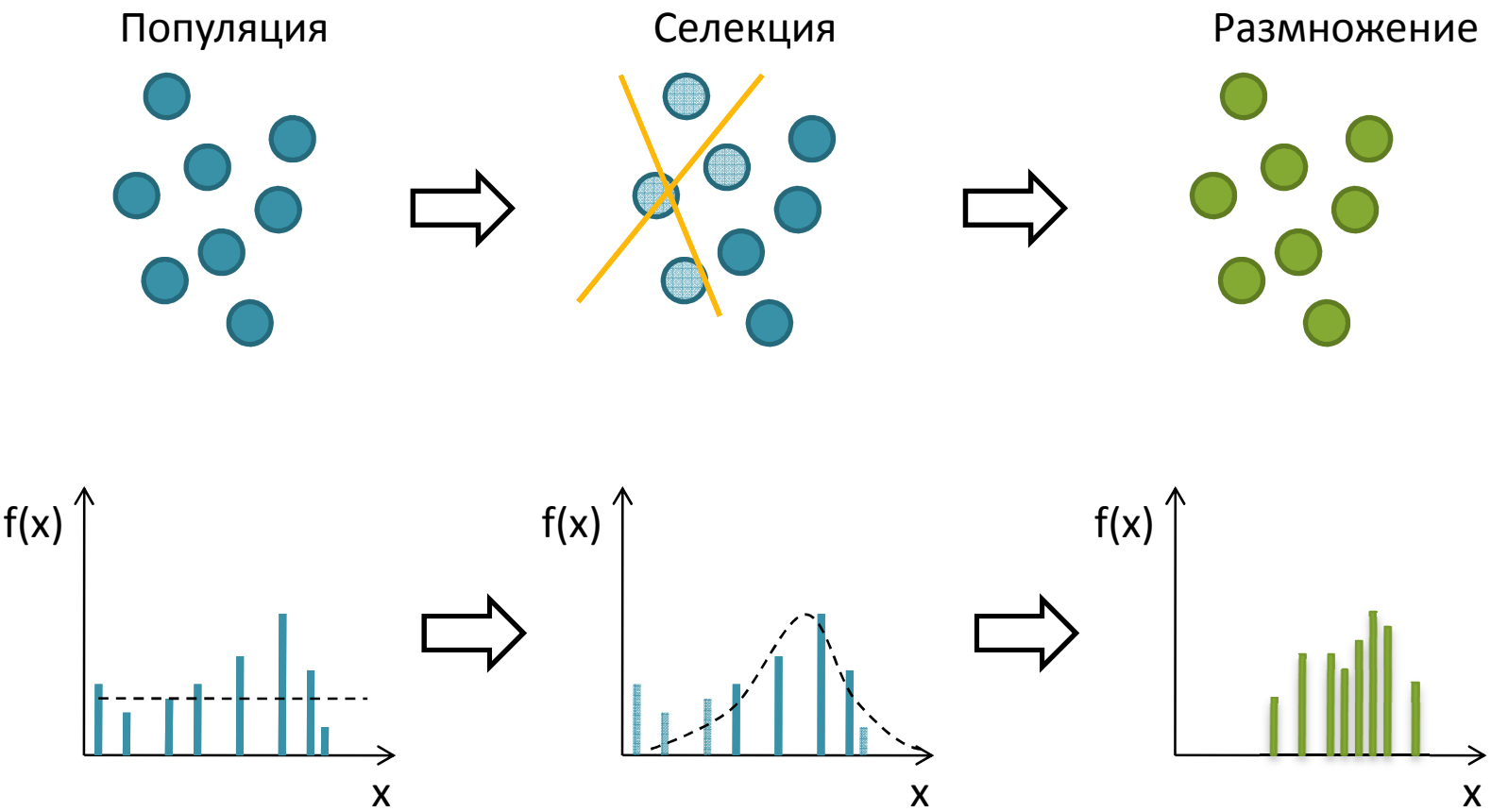
Вводная публикация: глава 5 «Оптимизация путем аппроксимации модели» из книги [Luke, 2009]. Перевод главы: http://qai.narod.ru/GA/meta-heuristics_9.pdf

Результаты BBOB-2009



Общая идея

1 мерный случай: f(x) → max



Общая схема

1. Начальная популяция генерируется из равномерного распределения P(0). t = 0.
2. Оценка приспособленности.
3. Селекция.
4. Обновление P(t) с использованием распределения P’(t), которое описывает приспособленных особей.
5. Если критерии останова не выполнены, то t = t+1 и переход на **Шаг 2**.

Предыстория

При работе с бинарными строками имеет смысл хранить информацию о вероятности 1 в каждой позиции [Baluja, 1994, Muhlenbein, Paas, 1996] вместо всей популяции.

0.84	0.53	0.09	. . .	0.35

Вероятности обновляются в зависимости от результатов селекции.

Позволяет избежать преждевременной сходимости.

Сокращение требований к памяти. [Sastry et al., 2007] Onemax для 2^{25} (> 33 млн.) разрядов.

Предыстория

- При решении задачи оптимизации логично учитывать взаимосвязи между переменными для повышения эффективности.
- Как их выявить? Linkage-learning: изучение зависимостей между «*строительными блоками*» для (предположительно) декомпозируемых задач. Исследования лаборатории Д. Голдберга с конца 80-х (!).

Предыстория

- ЭА сходится к одному решению по результатам запуска (если не применяться специальные приемы). Можно ли описать всю популяцию одним распределением? Да!

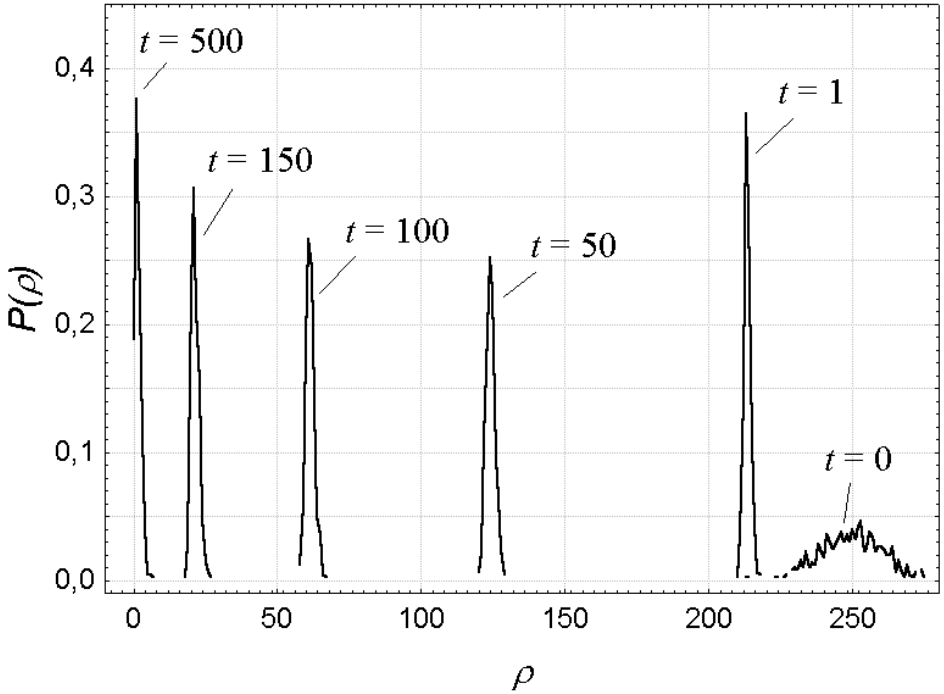


Рисунок из [Редько, Цой, 2005]

Предыстория

3 «В»:

- Вероятности вместо значений генов.
- Взаимосвязи между оптимизируемыми переменными.
- Возможность описания популяции одним распределением.

Авторы АОР

- Martin Pelikan

http://www.cs.umsl.edu/~pelikan/

Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)

University of Missouri in St. Louis

- Heinz Muehlenbein

http://muehlenbein.org/

Fraunhofer Institute AiS

Общая классификация АОР

- Без учета зависимостей (Univariate).
- Попарные зависимости (Bivariate).
- Множественные зависимости (Multivariate).

Одномерные АОР

Игнорируются зависимости между переменными → задача считается разделимой, т.е. можно проводить оптимизацию независимо для каждой переменной.

$$p(X_1, X_2, ..., X_n) = p(X_1)p(X_2)...p(X_n)$$

Для каждой переменной можно хранить одномерное распределение.

Популяция: {множество из *n* одномерных распределений}.

Population-Based Incremental Learning [Baluja, 1994]

- Бинарные переменные.
- Вектор вероятностей для единичных разрядов.

Population #1	Population #2	Population #3
0 0 1 1	1 0 1 0	1 0 1 0
1 1 0 0	1 1 0 0	0 1 0 1
1 1 0 0	1 1 0 0	1 0 1 0
0 0 1 1	1 1 0 0	0 1 0 1
Representation	**Representation**	**Representation**
0.5, 0.5, 0.5, 0.5	1.0 0.75 0.25 0.0	0.5 0.5 0.5 0.5

- Обновление вектора вероятностей по наиболее приспособленной особи.

P(t+1) = P(t)*a + ЛучшаяОсобь * (1-a).

- Используется мутация вектора вероятностей чтобы избежать преждевременной сходимости.

Univariate Marginal Distribution Algorithm (UMDA) [Muhlenbein and Paas, 1996]

- Для бинарных переменных.
- Рассматривается распределение $q(t)$ для каждой особи.
- В каждом поколении оценивается частота $r_{t;i}(x_i)$ появления i-го гена в популяции после селекции.
- Новая особь с хромосомой **x** генерируется в соответствии с оценкой вероятности:

$$q_{t+1}(\mathbf{x}) = \prod_{i=1}^{n} r_{t;i}(x_i)$$

Компактный ГА (cGA) [Harik et al., 1997]

- Бинарные переменные.
- Вектор *p* вероятностей единичных разрядов.
- В каждом поколении генерируются и оцениваются 2 особи.
- Вектор вероятностей обновляется по хромосоме особи-победителя **x**.

$$p_i(t+1) = \begin{cases} p_i(t) + 1/n, & x_i = 1 \\ p_i(t) - 1/n, & x_i = 0 \end{cases}$$

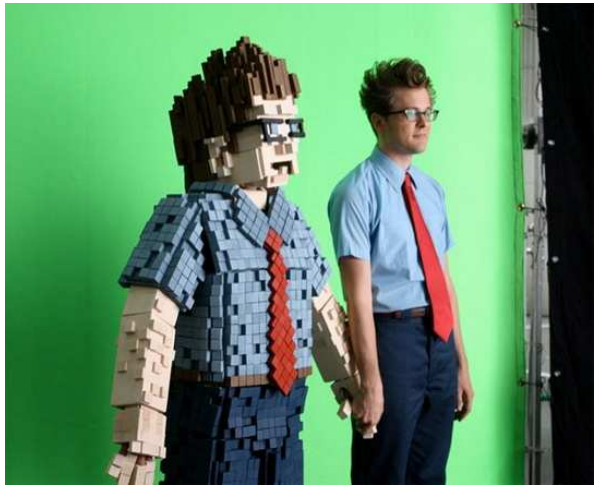
Вещественные переменные

Для бинарных и дискретных переменных построение оценок плотности распределения проще. Как быть с вещественными векторами?

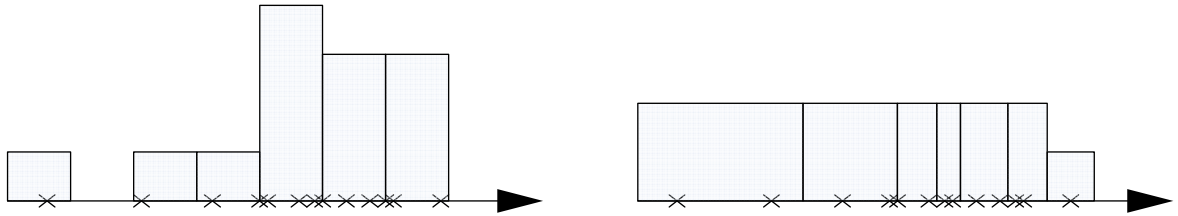
Два подхода:

- Дискретизация значений переменных.
- Использование непрерывных распределений и моделей (прямое представление).

Дискретизация



- Наивная
- Адаптивная
  - гистограмма с фиксированной высотой [Tsutsui et al., 2001]

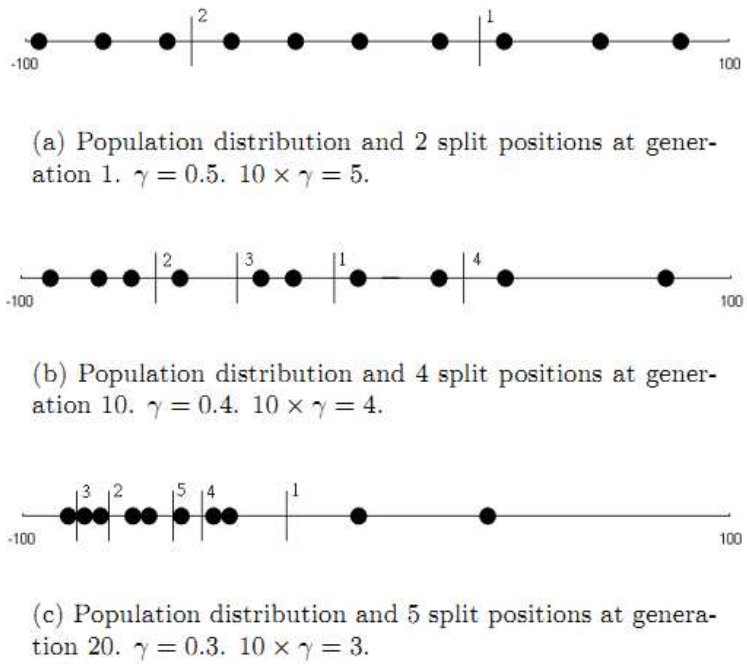


  - кластеризация значений переменной [Pelikan et al., 2003]
  - дискретизация «на лету» (split-on-demand (SoD)) [Chen et al., 2006]

Дискретизация

Дискретизация «на лету» [Chen et al., 2006]:

- Использовалась для алгоритма Extended Compact GA.
- Понятие «важности» интервала. Чем больше особей популяции «проецируется» на интервал, тем он важнее.
- Важные интервалы разбиваются на подинтервалы.



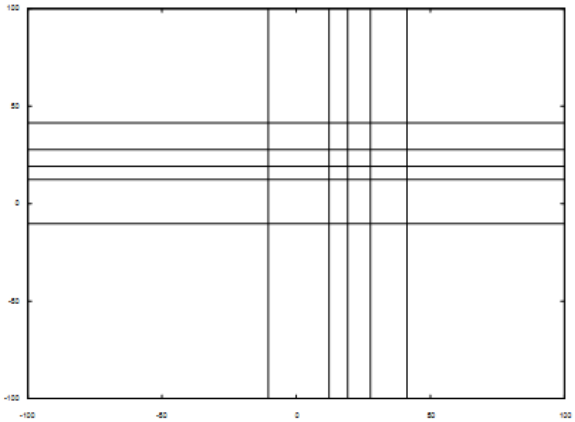
(a) Population distribution and 2 split positions at generation 1. $\gamma = 0.5$. $10 \times \gamma = 5$.

(b) Population distribution and 4 split positions at generation 10. $\gamma = 0.4$. $10 \times \gamma = 4$.

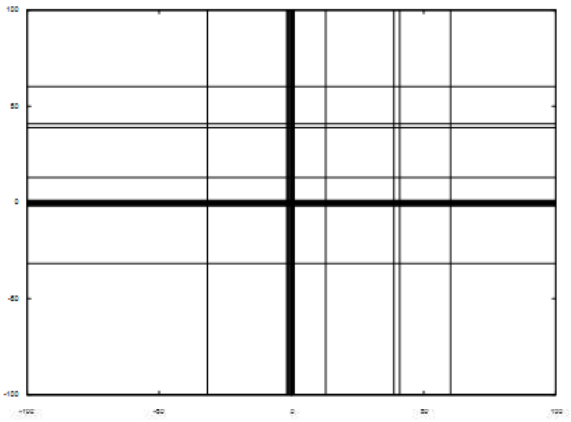
(c) Population distribution and 5 split positions at generation 20. $\gamma = 0.3$. $10 \times \gamma = 3$.

Рисунок из [Chen et al., 2006]

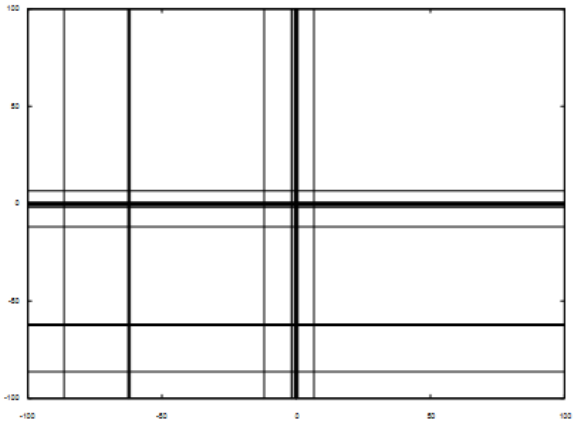
Дискретизация



(a) Generation 1.



(c) Generation 100.



(b) Generation 50.

Рисунок из [Chen et al., 2006]

Рассматривается сферическая функция:

$$f(\mathbf{x}) = \sum x_i^2$$

Прямое представление

The stochastic hill-climbing with learning by vectors of normal distributions (SHCLVND) [Rudlof, Koppen, 1996].

- Алгоритм локального поиска в $\mathbf{R}^n$ с подобием Хеббовского обучения.
- Каждая переменная описывается нормальным распределением.
- Обновление параметров распределений:

$$\vec{\mu} = \vec{\mu} + \mu_{move}(\vec{b}_{middle} - \vec{\mu}) = (1 - \mu_{move})\vec{\mu} + \mu_{move}\vec{b}_{middle}$$

$$\vec{\sigma} = \vec{\sigma}\sigma_{reduce}$$

Прямое представление

Непрерывный PBIL ($PBIL_C$) с использованием гауссовского распределения [Sebag and Ducoulombier, 1998].

- Возможность независимого обновления дисперсий для распределений переменных.
- Рассматривается несколько эвристик для независимого обновления дисперсий для разных переменных, побеждает следующая:

$$\sigma_i^{t+1} = (1-\alpha)\sigma_i^t + \alpha\sqrt{\frac{\sum_{j=1}^{K}(X_i^j - \overline{X}_i)^2}{K}} = (1-\alpha)\sigma_i^t + \alpha\sigma_{best,K}$$

Двумерные АОР

- Парные зависимости между переменными. Традиционная (не очень удачная) запись

$$p(X) = p(X_{i_1} \mid X_{i_2}) p(X_{i_2} \mid X_{i_3}) \ldots p(X_{i_{n-1}} \mid X_{i_n}) p(X_{i_n})$$

$i_1, \ldots i_n$ - перестановка множества $\{1,2,\ldots,n\}$

- Цепочечные и дерево-подобные структуры зависимостей.

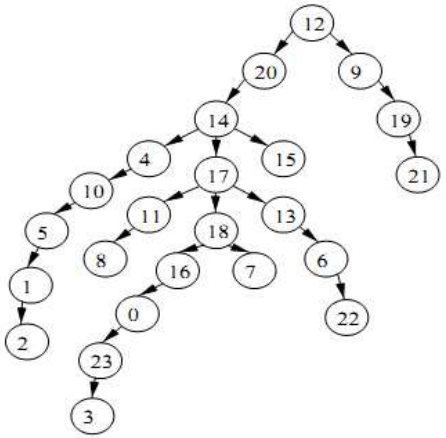


Рисунок из [Baluja, Davies, 1997]

Mutual-Information-Maximizing Input Clustering (MIMIC) [De Bonet et al, 1997]

- Работает с бинарными переменными.
- Равномерная функция плотности распределения:

$$p^{\theta}(x) = \begin{cases} \text{постоянная величина}, & \text{если } f(x) \le \theta \\ 0, & \text{в противном случае} \end{cases}$$

- Обновление оценки функции плотности распределения по медиане приспособленности.
- В популяции остаются только особи, приспособленность которых лучше некоторого порога.
- Два предположения:
  - $p^{\theta}(x)$ можно аппроксимировать, используя конечное множество данных.
  - Дивергенция Кульбака-Лейблера $D(p^{\theta-e}(x) \,\|\, p^{\theta}(x))$ мала при малых e.

Mutual-Information-Maximizing Input Clustering (MIMIC) [De Bonet et al, 1997]

- Целевая функция (h – энтропия):

$$f(\mathbf{X}) = h(x_{i_1} \mid x_{i_2}) + h(x_{i_2} \mid x_{i_3}) + \ldots + h(x_{i_{n-1}} \mid x_{i_n}) + h(x_{in})$$

- Используется жадный алгоритм для подбора оценки распределения $\hat{p}^{\theta}(x)$.
- Графическая модель для используемой функции плотности соответствует цепи.

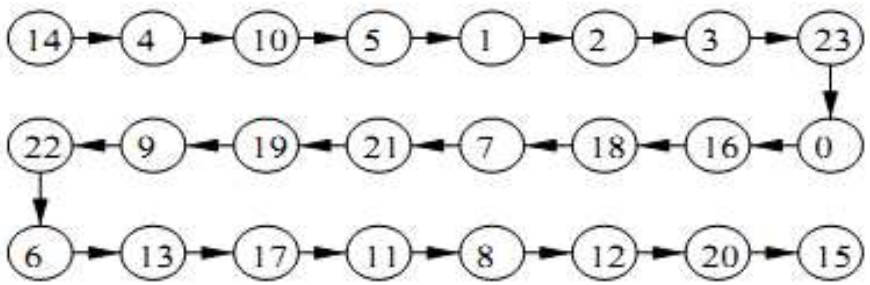
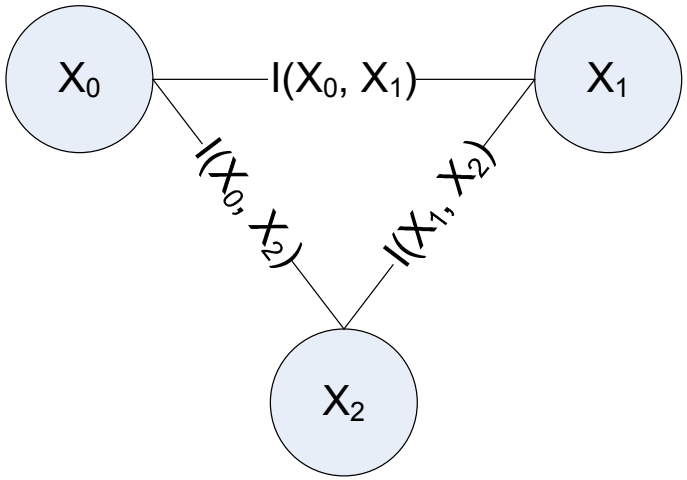


Рисунок из [Baluja, Davies, 1997]

Tree-based MIMIC [Baluja, Davies, 1997]

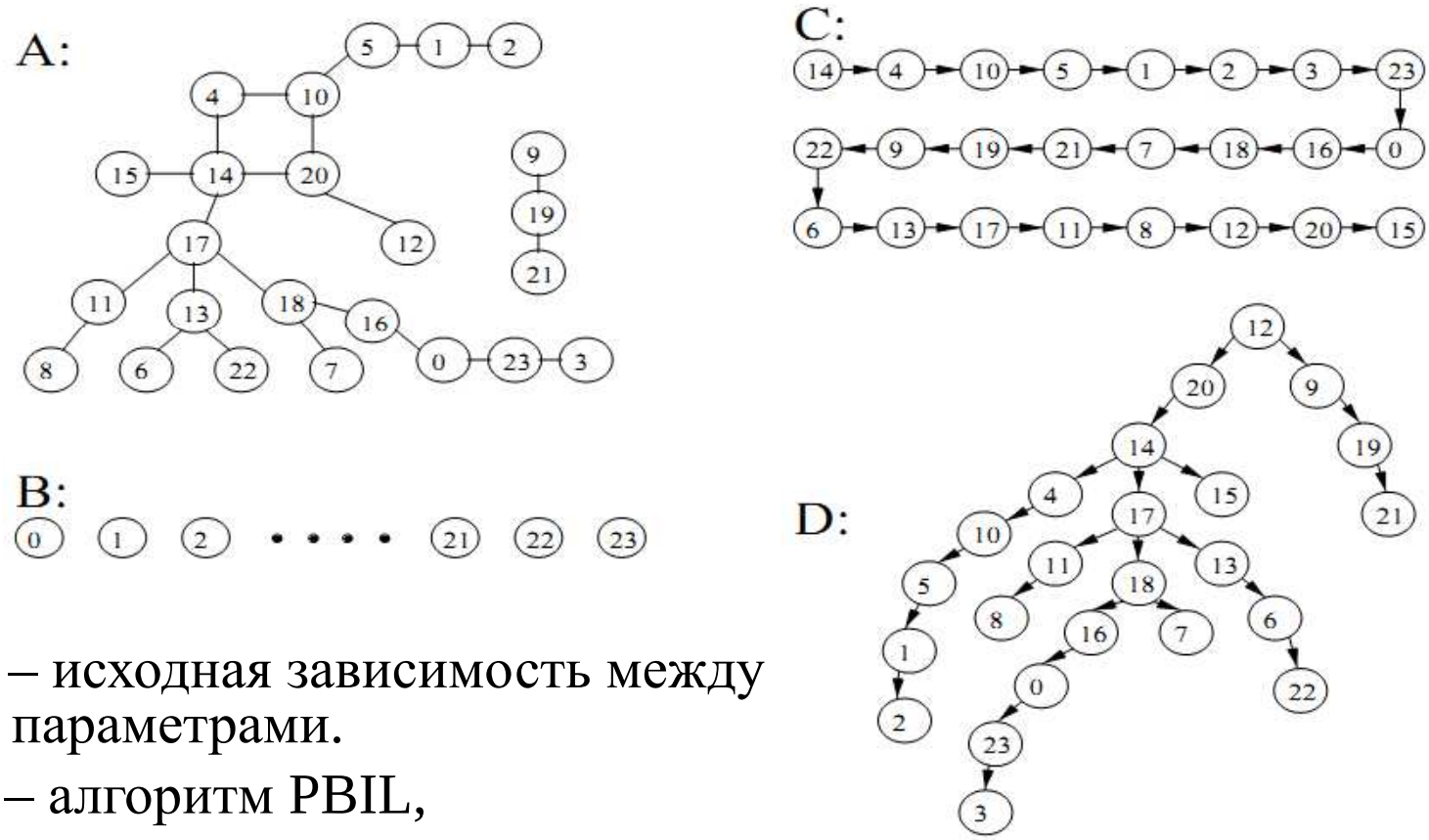
- Расширение алгоритма MIMIC.
- Работает с бинарными переменными.
- Другой выбор модели. Используется взвешенный неориентированный граф, вершины соответствуют переменным, ребра – **взаимной информации** между переменными. Для определения модели производится поиск **максимального связующего дерева**.



Tree-based MIMIC [Baluja, Davies, 1997]

- Для поиска дерева используется алгоритм из [Chou, Liu, 1968]
- В алгоритме в качестве «памяти» используются оценки $\mathbf{A}[x_j = \mathrm{a}, x_i = \mathrm{b}]$ количества «хороших» строк, содержащих комбинацию $x_j = \mathrm{a}$, $x_i = \mathrm{b}$ для всех пар (x_j, x_i).
- На каждой итерации производится «ослабление памяти»: умножение $\mathbf{A}$ на коэффициент < 1.

Tree-based MIMIC [Baluja, Davies, 1997]



A – исходная зависимость между параметрами.

B – алгоритм PBIL,

C – алгоритм MIMIC,

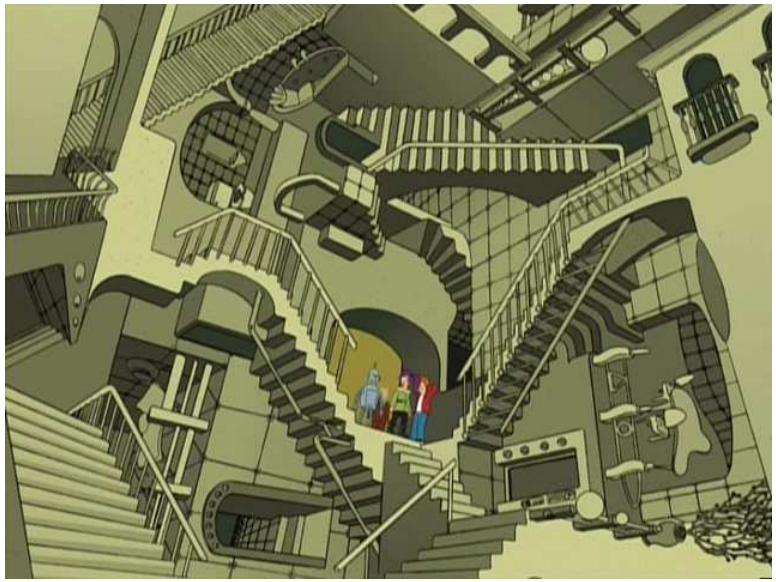
D – алгоритм Tree-based MIMIC.

Многомерные АОР

Вероятностная модель в наиболее общем виде:

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i \mid \Pi_i)$$

«родительские» переменные



Extended compact genetic algorithm (ECGA) [Harik, 1999]

Переменные разделяются на кластеры, которые полагаются независимыми.

1. Генерация случайной популяции из N особей.
2. Турнирная селекция с параметром t.
3. Построение вероятностной модели популяции с использованием жадного MPM поиска (кластеризация переменных).
4. Если модель сошлась – останов.
5. Генерация новой популяции с использованием сконструированной модели.
6. Переход на **Шаг 2**.

Covariance Matrix Adaptation ES (CMA-ES) [Hansen, 2011]

- Популяция аппроксимируется многомерным гауссовским распределением.
- Алгоритм зарождался как алгоритм с адаптивной подстройкой направления и шага мутации в эволюционных стратегиях.
- Основной подстраиваемый параметр: матрица ковариации (описывает взаимосвязи между переменными).
- Используется «траектория эволюции»

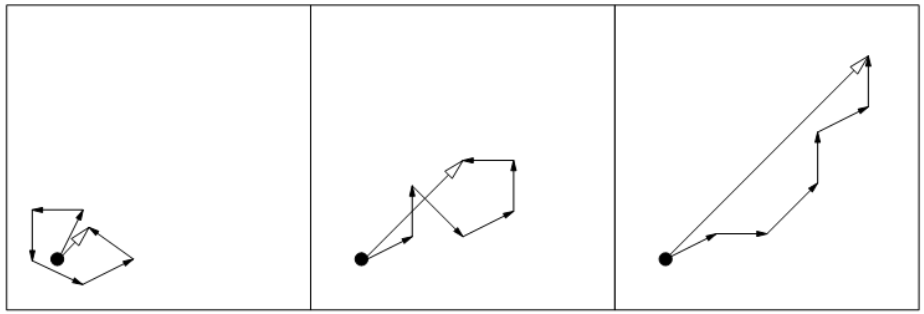


Рисунок из [Hansen, 2011]

Iterated Density Estimation Evolutionary Algorithm (IDEA) [Bosman,Thierens 1999]

- Функция плотности распределения как в алгоритме MIMIC:

$$p^{\theta}(x) = \begin{cases} \text{постоянная величина}, & \text{если } f(x) \le \theta \\ 0, & \text{в противном случае} \end{cases}$$

- Для дискретных переменных применяется оценка вероятности попадания в ячейки (bins).
- Для непрерывной оптимизации функция плотности аппроксимируется с использованием смеси гауссианов.

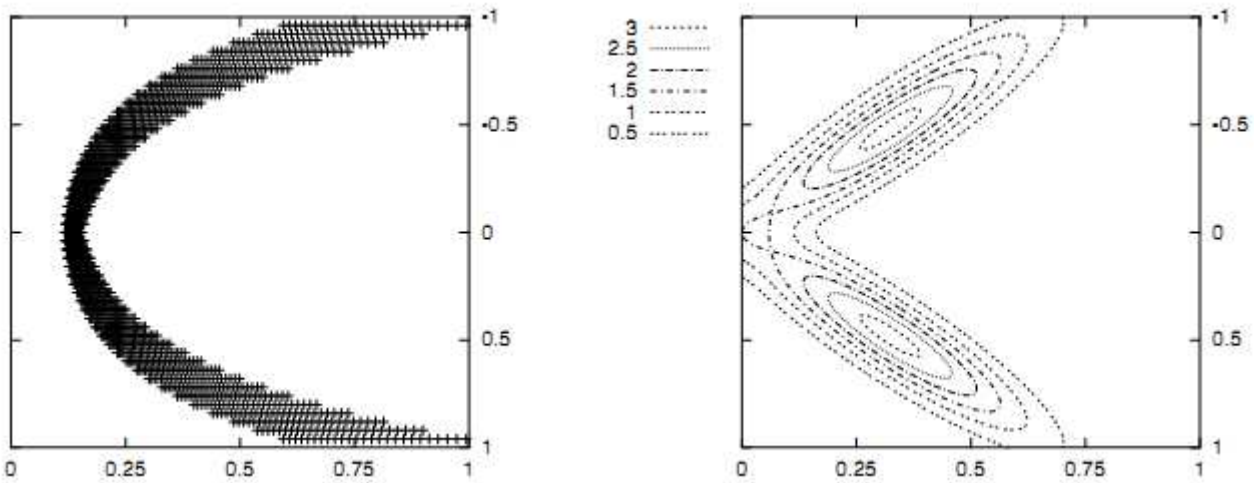
Iterated Density Estimation Evolutionary Algorithm (IDEA) [Bosman,Thierens 1999]

1. Инициализация.
2. Оценка популяции.
3. Селекция.
4. Оценка параметров распределения.
5. Генерация потомков.
6. Обновление популяции с использованием родительских особей и потомков.
7. Оценка популяции.
8. Если критерий останова не выполнен, то переход на **Шаг 3**.

Mixed IDEA (mIDEA) [Bosman, Thierens, 2001]

- Производится кластеризация особей.
- Для каждого кластера адаптируется свое распределение.
- Результирующее распределение вычисляется как взвешенная сумма распределений для кластеров.
- Для определения связей между переменными используется байесовский информационный критерий (BIC, критерий Шварца).

Рисунок из [Bosman, Thierens, 2001]



AMaLGaM-IDEA [Bosman et al., 2008] и iAMaLGaM-IDEA [Bosman et al., 2009]

- В качестве оценки распределения используется многомерный гауссиан, параметры которого определяются по принципу максимального правдоподобия.
- Масштабирование матрицы ковариации для избежания преждевременной сходимости.
- Прогноз для смещения центра распределения на следующем этапе.
- [iAMaLGaM] вычисление распределения для популяции производится с учетом параметров распределения на предыдущих этапах.

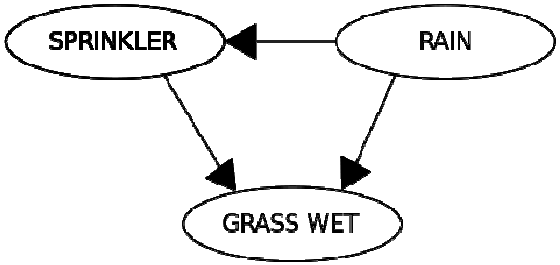
Алгоритм байесовской оптимизации (АБО) [Pelikan et al., 2000]

Bayesian optimization algorithm (BOA)

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i \mid \Pi_i)$$

Для построения моделей зависимостей между переменными используются сети доверия.

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Рисунок:
http://ru.wikipedia.org/wiki/Файл:SimpleBayesNet.svg

Алгоритм байесовской оптимизации (АБО) [Pelikan et al., 2000]

Построение модели:

1. Начинаем с сети без ребер.
2. Добавляется по 1 ребру. Выбор добавляемого ребра осуществляется с использованием метрики Байеса-Дирихле [Heckerman et al., 1995] (чем больше, тем лучше).
3. Во избежание получения слишком сложной модели используется ограничение сверху на число родительских вершин.

Можно использовать BIC [Etxeberria, Larranaga, 1999, Muhlenbein, Mahnig, 1999]

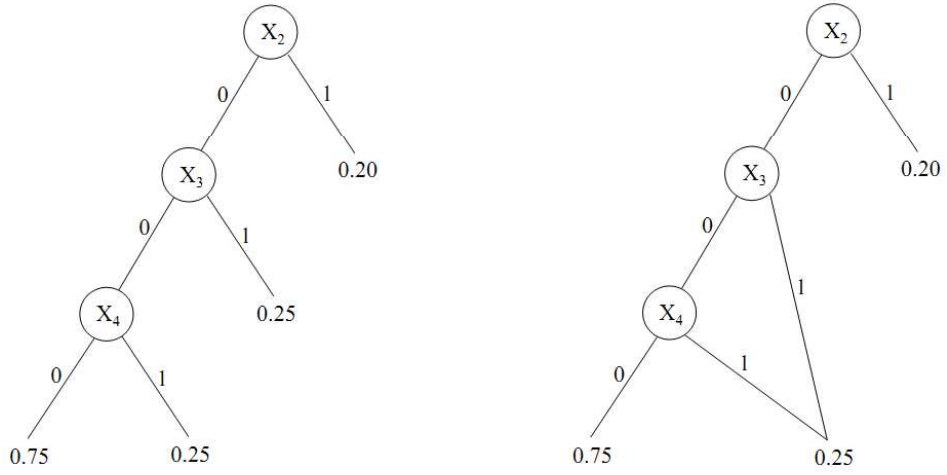
Real-coded Bayesian optimization algorithm (rBOA) [Ahn et al., 2004]

- Декомпозиция структурной модели (графа) → множество подмоделей.
- Для каждой подмодели производится кластеризация отобранных особей и для каждого кластера ищется гауссиан.
- Каждая подмодель описывается взвешенной суммой гауссианов.
- Новая особь генерируется в виде «конкатенации» экземлпяров из подмоделей.

Алгоритм иерархической Байесовской оптимизации [Pelikan, 2002]

Hierarchical Bayesian optimization algorithm (hBOA).

- Используются байесовские сети с решающими графами для декомпозиции задачи.



(a) Decision tree. (b) Decision graph.

- Restricted tournament replacement (RTR) [Harik, 1995] – конкуренция происходит только между похожими особями.

Другие варианты

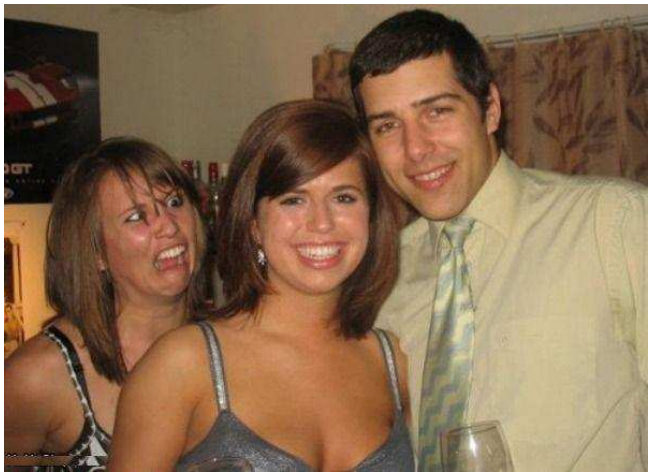
- Инкрементные АОР.
- Многоэкстремальные АОР.
- АОР с интеграцией экспертных знаний.
- Использование цепей Маркова.
- АОР для генетического программирования (кодирование деревьев и грамматик).
- АОР для кодирования перестановок.

Преимущества АОР

- Адаптация операторов к структуре рассматриваемой задачи.
- Выявление структуры задачи.
- Использование априорной информации.
- Сокращение требований к потребляемой памяти.

Недостатки АОР

- Большая вычислительная сложность.
- В ряде случаев может потребоваться больший объем оперативной памяти, чем при использовании «традиционного» ЭА.



Применения АОР

По данным статьи [Hauschild, Pelikan, 2011]:

- Структурный синтез антенн.
- Многокритериальная задача об упаковке.
- Проектирование плана восстановительных работ.
- Сокращение алфавита аминокислот для прогноза структуры протеина.
- Идентификация кластеров генов со схожими профилями экспрессии.
- Лесное хозяйство.
- Управление портфолио.
- Оптимизации химиотерапии для лечения рака.
- Проектирование сети для мониторинга состояния окружающей среды.

Дальнейшее развитие АОР [Hauschild, Pelikan, 2011]

- Параллелизация
- Аппроксимация целевой функции
- Гибридизация
- Оптимизация времени исполнения
- Однократное и инкрементное построение модели
- Встраивание проблемно-зависимой информации и экспертных знаний

Software & Open Source [Pelikan, 2004]

- BOA, BOA with decision graphs, dependency-tree EDA

http://medal.cs.umsl.edu/

- ECGA, BOA, and BOA with decision trees/graphs

http://www-illigal.ge.uiuc.edu/

- mBOA

http://jiri.ocenasek.com/

- PIPE

http://www.idsia.ch/~rafal/

- Real-coded BOA

http://www.evolution.re.kr/

- Demos of APS and EHBSA

http://www.hannan-u.ac.jp/~tsutsui/research-e.html

Заключение

На мой взгляд, алгоритмы оценки распределений – результат «взросления» ЭВ, перехода от «романтического» периода исследования всего, «что движется», к более зрелому и основательному изучению самих алгоритмов и их совершенствованию на основе математических методов.

АОР дополняют и обогащают машинное обучение и эволюционные вычисления.

Разнообразие существующих АОР позволяет применить их к решению самых разных задач оптимизации и поиска.

Список источников

1. [**Baluja, 1994**] Baluja Shumeet (1994), "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning", *Technical Report* (Pittsburgh, PA: Carnegie Mellon University) (CMU–CS–94–163).
2. [**Baluja, Davies, 1997**] Baluja, S., & Davies, S. (1997). Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. Proceedings of the International Conference on Machine Learning, 30–38.
3. [**Bosman et al., 2008**] Bosman P. A. N., J. Grahl, and D. Thierens. Enhancing the performance of maximum–likelihood Gaussian EDAs using anticipated mean shift. In G. Rudolph et al., editors, Parallel Problem Solving from Nature — PPSN X, pages 133–134, Berlin, 2008. Springer–Verlag.
4. [**Bosman et al., 2009**] Bosman P. A. N., Grahl J., Thierens D. AMaLGaM IDEAs in Noiseless Black-Box Optimization Benchmarking // Proceedings of Genetic and Evolutionary Computation Conference 2009, GECCO'09, July 8–12, 2009. P. 2247-2254.
5. [**Bosman, Thierens, 1999**] Bosman P.A.N., Thierens D. An algorithmic framework for density estimation based evolutionary algorithms. Utrecht University Technical Report UU–CS–1999–46, 1999. ftp://ftp.cs.uu.nl/pub/RUU/CS/techreps/CS-1999/1999-46.ps.gz
6. [**Bosman, Thierens, 2001**] Bosman, P. A., & Thierens, D. (2001a). Advancing continuous ideas with mixture distributions and factorization selection metrics. In Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at the Genetic and Evolutionary Computation Conference GECCO2001 (pp. 208–212). Morgan Kaufmann.

Список источников

7. [**Chen et al., 2006**] Chen, C.-H., Liu, W.-N., & Chen, Y.-P. (2006). Adaptive discretization for probabilistic model building genetic algorithms. In Proceedings of the 8th annual conference on Genetic and evolutionary computation, GECCO ’06 (pp. 1103–1110). New York, NY, USA: ACM.
8. [**Chou, Liu, 1968**] Chou C., Liu, C. (1968) Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory, 14:462-467.
9. [**Etxeberria, Larranaga, 1999**] Etxeberria, R., & Larra˜naga, P. (1999a). Global optimization using Bayesian networks. Second Symposium on Artificial Intelligence (CIMAF-99), 332–339.
10. [**Hansen, 2011**] Hansen N. The CMA Evolution Strategy: A Tutorial. June 28, 2011.
11. [**Harik, 1995**] Harik, G. R. (1995). Finding multimodal solutions using restricted tournament selection. International Conference on Genetic Algorithms (ICGA-95), 24–31.
12. [**Harik et al., 1997**] Harik, G. R., Lobo, F. G., & Goldberg, D. E. (1997). The compact genetic algorithm (IlliGAL Report No. 97006). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.
13. [**Muehlenbein, Mahnig, 1999**] Muhlenbein, H., & Mahnig, T. (1999). FDA – A scalable evolutionary algorithm for the optimization of additively decomposed functions. Evolutionary Computation, 7(4), 353–376.
14. [**Muehlenbein, Paas, 1996]** Muhlenbein H., Paas G. From recombination of genes to the estimation of distributions I. Binary parameters. Parallel Problem Solving from Nature, 1996, pp. 178–187.

Список источников

15. [**Pelikan, 2002**] Martin Pelikan (2002). *Bayesian optimization algorithm: From single level to hierarchy*. Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, IL. Also IlliGAL Report No. 2002023.
16. [**Pelikan, 2004**] Martin Pelikan, Probabilistic Model-Building GAs. http://medal.cs.umsl.edu/files/PMBGA_tutorial.pdf
17. [**Pelikan et al., 2000**] Pelikan M., Goldberg D.E., Cantu-Paz E. (2000). Linkage problem, distribution estimation, and Bayesian networks. Evolutionary Computation, 8(3), 311–341. Also IlliGAL Report No. 98013.
18. [**Pelikan et al., 2003**] Pelikan, M., Goldberg, D. E., & Tsutsui, S. (2003). Getting the best of both worlds: Discrete and continuous genetic and evolutionary algorithms in concert. Information Sciences, 156(3-4), 147 – 171. Evolutionary Computation.
19. [**Rudlof, Koppen, 1996**] Rudlof S., Koppen, M. (1996). Stochastic hill climbing with learning by vectors of normal distributions. Nagoya, Japan
20. [**Sebag, Ducoulombier, 1998**] Sebag, M., & Ducoulombier, A. (1998). Extending population-based incremental learning to continuous search spaces. In Parallel Problem Solving from Nature (pp. 418–427). Berlin. Heidelberg: Springer Verlag.
21. [**Tsutsui et al., 2001**] Tsutsui, S., Pelikan, M., & Goldberg, D. E. (2001). Evolutionary algorithm using marginal histogram models in continuous domain. Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), 230–233.
22. [**Редько, Цой, 2005**] Редько В.Г., Цой Ю.Р. Оценка эффективности эволюционных алгоритмов // Доклады АН, 2005. Т. 404. № 3. С. 312-315.

Цой Ю. Алгоритмы оценки распределений и байесовской оптимизации

СПАСИБО ЗА ВНИМАНИЕ!