

Реализация и использование эволюционных алгоритмов для минимизации функций в ROOT

Михаил Жабицкий

Объединенный институт ядерных исследований, Дубна

MPAMCS 2014, 25 августа, 2014, Дубна

Содержание

- 1 Минимизация функции вещественных переменных
 - Поиск глобального минимума
 - Асинхронная дифференциальная эволюция
 - АДЭ с адаптивной корреляционной матрицей
- 2 Программная реализация алгоритма
 - Параллельные расчеты
 - Интеграция с ROOT
 - Примеры решаемых задач
- 3 Результаты

Поиск глобального минимума

Поиск вектора $\mathbf{x}^* = \{x_j\}_{j=0, \dots, D-1}$, минимизирующего целевую функцию $f(\mathbf{x})$:

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega,$$

где $\Omega = R^D$ — область поиска.

«Отягчающие» особенности некоторых задач:

- Ограничения на параметры $\varphi(\mathbf{x}) < 0$
- Многопараметрические задачи $D = 10 \dots 100$
- Многомодальные целевые функции
- Производные целевой функции недоступны
- Целевые функции, требующие значительных вычислений

Дифференциальная эволюция

- Эволюционный алгоритм (ЭА) — алгоритм, реализующий процессы, аналогичные **отбору**, **мутации** и **скрещиванию** в биологии
- Дифференциальная эволюция (ДЭ) — эволюционный алгоритм со специфическим оператором **мутации**
- Разработан Р. Шторном и К.В. Прайсом в 1995
[K. Price, R. Storn// J. Global of Optimization 11 (1997) 341]
- Оперирует **популяцией** векторов размером N_p
- Каждый член популяции — вектор в пространстве параметров $\Omega = R^D$

[K. Price, R. Storn, J.A. Lampinen "Differential evolution — A Practical Approach to Global Optimization", Springer, 2005]

[S. Das, P.N. Suganthan// IEEE Trans. Evol. Comp. 15 (2011) 4]

Асинхронная Дифференциальная эволюция

```
// инициализация популяции  $\{\mathbf{x}_i\}_{i=0,\dots,N_p-1}$ ,  $\mathbf{x}_i = \{x_{i,j}\}_{j=0,\dots,D-1}$ 
do {
   $i = \text{choose\_target\_vector}()$ ; // выбор целевого вектора  $i$ 
  // Мутация:
   $\mathbf{v}_i = \mathbf{x}_i + F(\mathbf{x}_p - \mathbf{x}_q)$ ; // мут. вектор,  $r \neq p \neq q$  — сл. индексы
  // Кроссовер (рекомбинация):
  for ( $j = 0$ ;  $j < D$ ;  $j = j + 1$ )
     $u_{i,j} = \begin{cases} v_{i,j} & \text{rand}(0,1) < C_r \text{ or } j = j_{\text{rand}} \\ x_{i,j} & \text{иначе} \end{cases}$  // пробный вектор

  // Отбор:
  if ( $f(\mathbf{u}_i) < f(\mathbf{x}_i)$ )
     $\mathbf{x}_i = \mathbf{u}_i$ ;
} while (пока не выполнен критерий останова);
```

[E. Zhabitskaya, M. Zhabitsky // LNCS 7125, 328, 2012]

Асинхронная ДЭ: классификация алгоритмов

мутация:
$$\mathbf{v}_i = \mathbf{x}_r + F(\mathbf{x}_p - \mathbf{x}_q)$$

\mathbf{x}_i — целевой вектор;

\mathbf{x}_r — базовый вектор

$(\mathbf{x}_p - \mathbf{x}_q)$ — дифференциальный вектор

DE/w/x/y/z в соответствии с операторами Мутации и Кроссовера:

w отвечает заменяемому целевому вектору;

x отвечает базовому вектору;

y — число дифференциальных векторов;

z — тип кроссовера (биномиальный или экспоненциальный).

rand/rand/1/bin
$$\mathbf{v}_{\text{rand}} = \mathbf{x}_{\text{rand}} + F(\mathbf{x}_p - \mathbf{x}_q)$$

rand/best/1/bin
$$\mathbf{v}_{\text{rand}} = \mathbf{x}_{\text{best}} + F(\mathbf{x}_p - \mathbf{x}_q)$$

worst/best/bin
$$\mathbf{v}_{\text{worst}} = \mathbf{x}_{\text{best}} + F(\mathbf{x}_p - \mathbf{x}_q)$$

Равномерное (биномиальное) скрещивание с фиксированным C_r

Покоординатная комбинация **мутантного** вектора \mathbf{v} с **целевым** вектором \mathbf{x} для создания **пробного** вектора \mathbf{u} :

$$u_{i,j} = \begin{cases} v_{i,j} & \text{rand}(0, 1) < C_r \text{ или } j = j_{\text{rand}} \\ x_{i,j} & \text{в противном случае.} \end{cases}$$

Результат итераций Дифференциальной эволюции критически зависит от **вероятности скрещивания** C_r :

$C_r = 0$ для разделяемых задач

$C_r = 1$ для неразделяемых задач (инвариантность отн. вращений)

$C_r = 0.9$ — типичный начальный выбор

АДЭ с адаптивной корреляционной матрицей

[E. Zhabitskaya, M. Zhabitsky, GECCO-2013, 455]

Популяция ДЭ адаптируется с рельефу минимизируемой функции:

[K. Price, R. Storn, J.A. Lampinen, Springer, 2005]

Популяцию можно рассматривать как выборку для определения парных корреляций между параметрами:

$$q_{jk} = \frac{1}{N_p-1} \sum_{i=0}^{N_p-1} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k),$$
$$s_{jk} = \frac{q_{jk}}{\sqrt{q_{jj}q_{kk}}},$$

Адаптивная корреляционная матрица:

$$C' = (1 - c)C + cS.$$

[A. Auger, N. Hansen, CMA-ES, IEEE CEC (2005) 1769]

АДЭ с адаптивной корреляционной матрицей (II)

Пример: композитная функция

$$f = f_{\text{Rosenbrock}}(x_0, x_1) + f_{\text{Rosenbrock}}(x_2, x_3)$$

$$C = \begin{pmatrix} C_{00} & 0.910 & 0.027 & 0.038 \\ 0.910 & C_{11} & 0.016 & 0.015 \\ 0.027 & 0.016 & C_{22} & 0.945 \\ 0.038 & 0.015 & 0.945 & C_{33} \end{pmatrix}$$

[E. Zhabitskaya, M. Zhabitsky, GECCO-2013, 455]

АДЭ с адаптивной корреляционной матрицей (III)

Выбор координаты m и порога c_{thr} :

$$m = [N_p \text{rand}(0, 1)], \quad c_{\text{thr}} = \text{rand}(0, 1);$$

$$\{I_m\} = \forall j : |c_{mj}| > c_{\text{thr}}.$$

$$C = \left(\begin{array}{cc|cc} C_{00} & 0.910 & 0.027 & 0.038 \\ 0.910 & C_{11} & 0.016 & 0.015 \\ \hline 0.027 & 0.016 & \mathbf{C_{22}} & 0.945 \\ 0.038 & 0.015 & \mathbf{0.945} & C_{33} \end{array} \right)$$

$$u_{i,j} = \begin{cases} v_{i,j} & \text{если } j \in \{I_m\} \\ x_{i,j} & \text{в противном случае.} \end{cases} \Rightarrow \text{Скращивание в выделенном подпространстве}$$

[E. Zhabitskaya, M. Zhabitsky, GECCO-2013, 455]

Адаптация размера популяции N_p

	малый N_p	большой N_p
Вероятность сходимости	—	+
Скорость сходимости	+	—

Алгоритм АДЭ с рестартом [Zh & Zh// LNCS 8236 (2013) 555]:

- Малый размер начальной популяции N_p
- Рестарт с увеличенным $N_p \leftarrow kN_p$, если диагностирована стагнация сходимости (длительное отсутствие прогресса или вырождение популяции)

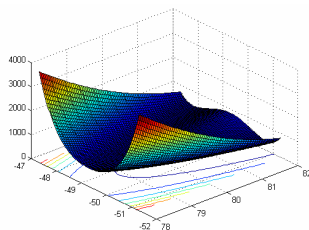
Адаптация размера популяции в соответствии со сложностью решаемой проблемы!

Частично разделяемые задачи

Смещенная функция Розенброка

$$f_6(\mathbf{x}) = \sum_{j=1}^{D-1} \left(100 (z_j^2 - z_{j+1}^2) + (z_j - 1)^2 \right) + f_{\text{bias}},$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o} + 1, \quad \mathbf{x} \in [-100, 100]^D$$



Критерии оценки

- Вероятность сходимости

$$P_{\text{succ}} = \frac{N_{\text{succ}}}{N_{\text{trials}} = 25}$$

- Кол-во вызовов f (усредненное по сошедшимся попыткам)

$$\langle N_{\text{feval}} \rangle = \frac{1}{N_{\text{succ}}} \sum_i (N_{\text{feval}})_i$$

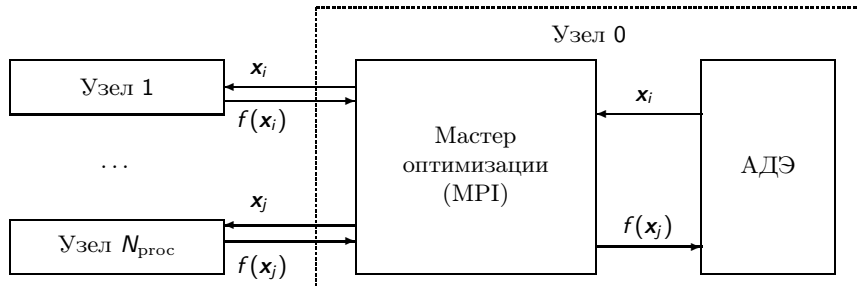
Частично разделяемые задачи

$$D = 20 : \quad f(\mathbf{x}) = f_{\text{Ros}}(x_0, x_1) + f_{\text{Ros}}(x_2, \dots, x_4) + \\
 f_{\text{Ros}}(x_5, \dots, x_8) + f_{\text{Ros}}(x_9, \dots, x_{13}) + f_{\text{Ros}}(x_{14}, \dots, x_{19})$$

DE/rand/rand/1 стратегия:

	C_r	P_{succ}	mean	median	std.dev
bin	0	0	—	—	—
bin	0.9	1	4.93e+05	4.14e+05	1.95e+05
bin	JADE	0.44	1.19e+07	—	1.11e+06
acm	—	1	6.82e+04	6.57e+04	1.74e+04

Параллельные расчеты на распределенных вычислительных системах (АДЭ)



- Master/Slave модель
- Полная и эффективная! загрузка вычислительных узлов
- Мастер оптимизации в стандарте MPI (MPICH2)

[Е. Жабицкая, М. Жабицкий, Мат. моделирование 24 (2012) 33]

Интеграция с ROOT

- **ROOT** (<http://root.cern.ch>) — основной пакет анализа экспериментальных данных в физике частиц
- **MINUIT** — программа минимизации функций вещественных переменных
[F. James, CERN, 1972]
- **Migrad** — реализация квазиньютоновского алгоритма, основанного на формуле Давидона-Флетчера-Пауэлла
- В ROOT реализованы 2 варианта:
MINUIT — f2c конверсия оригинального FORTRAN MINUIT
MINUIT2 — C++ версия MINUIT
- **ROOT::Math::Minimizer** — базовый класс для оптимизационных алгоритмов в ROOT

Квазиньютоновский метод vs Эволюционный алгоритм

По функциональности:

Квазиньютоновский метод	Эволюционный алгоритм
Построение квадратичной формы (ряд Тейлора)	Без производных
Целевая функция	
Дифференцируема	\forall , в т.ч. разрывные
Искомый минимум	
Локальный	Глобальный

С точки зрения пользователя:

Квазиньютоновский метод	Эволюционный алгоритм
Начальная точка	Начальная область
Начальная оценка шага	—
Минимизируемая целевая функция	
Минимум функции	

Интеграция Эволюционных алгоритмов в ROOT

- Minimizer в ROOT инициализируется в скрытом уровне
- Методы для задания начальной области поиска
- Интерфейс для задания внутренних параметров минимизатора
- Plugin для инициализации минимизатора
- Внешняя библиотека реализации алгоритма

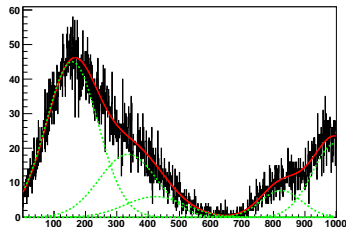
git.cern.ch

Примеры решаемых задач

Приложения в ядерной физике и физике конденсированных сред
 [Евгения Жабицкая, МРАМCS, 29.08.2014]

Анализ спектра без целеуказания:

Alg.	N_{peaks}	D	P_{succ}	median
Migrad	2	6	0.43	0.6e+3
ADE	2	6	0.90	1.8e+3
Migrad	5	15	0.32	2.8e+3
ADE	5	15	0.81	4.0e+4



Результаты

- Разработан алгоритм Асинхронной Дифференциальной эволюции с адаптивной корреляционной матрицей
- Алгоритм эффективен для нахождения минимума частично-разделяемых задач
- Реализована автоматическая адаптация параметров алгоритма (не требует доп. усилий от пользователя)
- Интеграция с пакетом ROOT для решения оптимизационных задач