

Автоматический выбор параметров алгоритмов
Карпов Пётр Михайлович

Automatic selection of algorithm parameters
Peter Karpov

Website: inversed.ru
Email: PeterKarpov@inversed.ru

2012.07.04

Аннотация

В работе рассмотрена проблема автоматической настройки параметров алгоритмов и оценки их влияния. Предложен общий метод её решения в условиях ограниченных вычислительных ресурсов. Эффективность метода продемонстрирована на примере настройки метаэвристического алгоритма оптимизации.

Abstract

Present work studies the problem of automatic algorithm parameter tuning and effect estimation. A general method of solving the problem under condition of limited computational resources is proposed. Efficiency of the method is demonstrated on a metaheuristic optimization algorithm.

Введение

При использовании какого-либо алгоритма весьма важным может являться выбор его свободных параметров. Во многих исследованиях этому вопросу уделяется недостаточно внимания: авторы зачастую ограничиваются тем, что приводят их значения без какого-либо обоснования. Ручная настройка параметров обычно является весьма трудоёмкой процедурой. В то же время многие популярные алгоритмы обладают большим количеством параметров, существенно влияющих на результаты. Отдельно следует выделить метаэвристические методы оптимизации, к числу которых принадлежит например Генетический Алгоритм [1]. Поэтому в последнее время наблюдается значительный интерес к систематическим и автоматизированным методам настройки параметров [2][3].

Набор N свободных параметров обозначим $x = \{x_1, \dots, x_N\}$. Параметры могут быть численными, либо категориальными. Будем считать, что численные параметры могут принимать конечное количество заранее выбранных значений. Такой подход потенциально ограничивает точность настройки, но в то же время значительно упрощает дальнейшие рассуждения.

Настраиваемый алгоритм будем считать стохастическим, то есть дающим различные результаты при каждом запуске. Большинство метаэвристических методов оптимизации являются стохастическими в этом смысле. Пусть для настраиваемого алгоритма введена мера качества даваемых им в результате одного запуска результатов $Q(x)$. В качестве целевой функции (ЦФ) используем среднее значение качества получаемых результатов: $f(x) = \langle Q(x) \rangle$. В зависимости от задачи можно использовать и другие статистические меры, например медиану. Так как в реальности доступно лишь конечное количество запусков алгоритма, целевая функция является лишь оценкой и обладает статистической погрешностью.

Задачу автоматической настройки таким образом определим как выбор набора параметров x , дающего наилучшее среднее качество результатов настраиваемого алгоритма. В

зависимости от конкретной задачи, это может означать максимизацию или минимизацию $f(x)$. Дополнительной задачей является оценка влияния различных параметров на результаты работы алгоритма.

Выбор методики

Распространённым подходом к подобному рода задачам оптимизации и анализа влияния параметров является планирование экспериментов (Design of Experiments, DoE) [4]. Согласно DoE, следует выбрать экспериментальный план, обеспечивающий наибольшую точность результатов. В терминах этого подхода параметры именованы факторами, а их значения — уровнями. Гораздо более простой методикой является изменение одного фактора за раз (one factor at a time, OFAT) и наблюдение последующих эффектов. Такая процедура состоит из следующих шагов:

инициализировать параметры случайными значениями
для каждого параметра
 перебрать все значения
 выбрать лучшее

Если количество возможных значений параметра i обозначить за L_i , OFAT требует $1 + \sum_{i=1}^N (L_i - 1)$ экспериментов. С точки зрения методов оптимизации, OFAT весьма близок к локальному поиску. Разница состоит в том, что первый проводит фиксированное число экспериментов, последний же работает до тех пор, пока изменение, приводящее к улучшению, не может более быть найдено.

Большинство современных авторов рекомендуют использовать DoE вместо OFAT, приводя в качестве преимуществ первого статистическую мощьность и способность оценки взаимодействий факторов [5]. Однако методология DoE обладает и недостатками. Наибольшее внимание в ней уделяется экспериментальным планам с двух- и трёхуровневыми факторами. Создание планов, сочетающих факторы с произвольным количеством уровней, является весьма трудной задачей. Будучи же построены, такие планы могут требовать многократно большего числа экспериментов в сравнении с OFAT. Это ограничение является столь серьёзным, что делает методы DoE неприменимыми к задаче настройки параметров в формулировке, используемой в данной работе. Кроме того, экспериментальное сравнение DoE с OFAT в задачах оптимизации [6] выявило, что DoE оказывается более эффективным лишь при выполнении специальных условий: случайные ошибки велики в сравнении с основными эффектами, а взаимодействие факторов является средним или слабым. В остальных случаях OFAT даёт лучшие результаты.

Предлагаемый метод

Из рассмотрения особенностей метода OFAT ясно, что его использование в качестве основы является вполне разумным. Однако его применение в чистом виде сопряжено с некоторыми трудностями. После одного прохода по всем параметрам нет никакой гарантии, что полученный результат является локально оптимальным, то есть следующий проход не даст более улучшений. Использование критерия останова в стиле локального поиска (продолжать пока есть улучшения) невозможно из-за того, что целевая функция может обладать значительной погрешностью, приводящей к сбоям в работе такого критерия. Можно было бы ввести фиксированное количество проходов, но это ставит нас перед дилеммой: малое их число приводит к потенциально субоптимальным результатам, большое — к увеличенному времени вычислений.

Сформулируем основные требования к искомому методу настройки параметров:

- *Локальная оптимальность*

Получаемые наборы параметров должны быть локально оптимальны с высокой вероятностью.

– *Толерантность к шуму*

Метод должен быть устойчив к наличию статистической погрешности целевой функции.

– *Статистическая эффективность*

Многие алгоритмы имеют высокую вычислительную стоимость. Метод настройки должен работать за разумное время, располагая лишь ограниченным числом тестовых запусков.

– *Простота*

Метод не должен содержать свободных параметров, ведь в противном случае придётся настраивать сам метод настройки.

Модификацией OFAT, соответствующей приведённым требованиям является итеративная схема с постепенным увеличением набираемой статистики:

инициализировать параметры случайными значениями

$M := 1$

повторять

 для каждого параметра

 для каждого уровня

 произвести M запусков алгоритма

 вычислить значение ЦФ

 выбрать лучший уровень

$M := k * M$

пока не вышло время

Количество запусков алгоритма на каждой итерации таким образом растёт в геометрической прогрессии. Такая схема имеет ряд преимуществ:

- Использование множества итераций сочетается с высокой точностью конечных результатов
- Каждая итерация использует хорошее начальное решение, полученное на предыдущей итерации
- Промежуточные результаты доступны в любой момент; чем дольше время работы, тем выше точность результатов
- Высокая статистическая погрешность на начальных стадиях ведёт к случайному блужданию, что может дать эффект глобального поиска

Метод однако наследует одну из слабостей OFAT – игнорирование взаимодействия факторов. При значительной степени такого взаимодействия полученное решение может сильно отличаться от глобально оптимального. Этот недостаток может быть устранён при помощи модификаций, направленных на улучшение качества глобального поиска, например использования популяции параметров.

Попробуем выбрать наилучший коэффициент прогрессии k . Допустим, что процесс настройки был прерван и выполнялась лишь часть t последней итерации. Вычислительные затраты полной последней итерации возьмём за единицу. Тогда суммарные затраты до момента прерывания составляют $k/(k-1) - 1 + t$. Максимальная погрешность ЦФ определяется количеством запусков на последней полностью завершённой итерации, то есть той, которая предшествовала прерванной. Для достижения более высокой точности необходимо максимизировать относительные вычислительные затраты на предпоследнюю итерацию. Они составляют

$$\eta(k, t) = \frac{k-1}{k(1+(k-1)t)}$$

Функция $\eta(k, t)$ достигает максимума при $k=1+1/\sqrt{t}$. В худшем случае «потерянной» является почти вся последняя итерация, то есть $t=1$, что даёт $k=2$. Если же положить время прерывания t случайной величиной, распределённой равномерно от 0 до 1, то после усреднения по t имеем:

$$\langle \eta(k) \rangle = \int_0^1 \eta(k, t) dt = \frac{\ln k}{k}$$

Эта функция соответствует относительным полезным затратам в среднем случае и достигает максимума при $k=e$.

Наконец, осталось определить порядок, в котором происходит настройка параметров. Порядок от наиболее важного к наименее важному параметру выглядит наилучшим по двум причинам. Во-первых, существенные изменения происходят как можно раньше, что положительно влияет на дальнейший процесс оптимизации. Во-вторых, после прерывания итерации в произвольный момент, самые важные параметры с высокой вероятностью уже подверглись настройке. Конечно, степень влияния параметров заранее неизвестна, но в следующей секции будет показано, что она может быть оценена непосредственно во время настройки.

Влияние параметров

Помимо нахождения оптимального набора параметров, важной задачей является оценка влияния каждого из них. Это оказывается возможным без каких-либо модификаций предложенного метода. Предположим, что поведение настраиваемого алгоритма описывается линейной моделью, то есть представимо в виде суммы вкладов отдельных факторов:

$$f(x) = b + \sum_{i=1}^N d(i, j),$$

где b — глобально оптимальное значение ЦФ, а $d(i, j)$ — влияние j -го значения параметра i . В случае, если все значения параметров выбирались бы случайно, среднее отклонение ЦФ от оптимального значения составило бы

$$S = \langle f(x) - b \rangle = \sum_{i=1}^N \langle d(i, j) \rangle$$

Относительный вклад каждого фактора в изменение ЦФ таким образом составляет

$$R_i = \frac{\langle d(i, j) \rangle}{S}$$

При настройке параметра i производится перебор его значений и вычисление массива ЦФ $g(i, j)$, $j=1..L_i$. Основываясь на линейной модели, влияние различных значений параметра i можно получить, вычтя из массива $g(i, j)$ его оптимум, то есть

$$d(i, j) = g(i, j) - \text{Opt}_{j=1}^{L_i} g(i, j),$$

где Opt означает минимум или максимум в зависимости от задачи.

Произведённый таким образом анализ чувствительности является локальным, то есть относится лишь к полученному набору параметров. Полученная информация оказывается полезной для задачи оптимизации. Как уже говорилось, настройку параметров выгодно осуществлять в порядке убывания важности. Таким образом, перед каждой итерацией следует производить сортировку параметров по степени влияния. Перед первой итерацией порядок можно выбрать произвольным образом.

Пример использования

Для проверки предложенного метода с его помощью была произведена настройка алгоритма роевого интеллекта — дискретного аналога алгоритма роя частиц [1]. Алгоритм

использовался для решения задачи коммивояжёра, состоящей в нахождении кратчайшего замкнутого маршрута, проходящего через заданные точки. Были использованы точки из файла lin105 библиотеки TSPLIB [7]. Алгоритм содержит 6 параметров (табл. 1). При таких условиях полный факторный эксперимент, то есть тестирование всех возможных сочетаний, требует примерно в 140 раз большего количества вычислений ЦФ, чем одна итерация предлагаемого метода.

Параметр	Количество возможных значений
Размер популяции	8
Сила притяжения	10
Зависимость притяжения от времени	7
3 других параметра	2 значения у каждого

Таблица 1: Параметры настраиваемого алгоритма.

Сравнение различных алгоритмов приведено в таблице 2. Из результатов тестирования следует, что автоматическая настройка алгоритма роевого интеллекта позволяет добиться очень высокой эффективности.

Алгоритм	Среднее относительное отклонение длины маршрута от оптимального значения, %
Роевой интеллект без настройки	11.8
Локальный поиск 2-opt [8]	10.0
Симуляция отжига [1]	1.7
Роевой интеллект после настройки	0.7

Таблица 2: сравнение эффективности алгоритмов в задаче коммивояжёра.

Заключение

В работе предложен метод автоматической настройки параметров алгоритмов. Такой метод избавляет пользователя от трудоёмкой ручной настройки и способен существенно повысить эффективность работы настраиваемого алгоритма. Вместе с выбором оптимального набора параметров осуществляется оценка влияния каждого из них. Метод обладает значительной общностью и нетребователен к количеству тестовых запусков алгоритма. Поддерживается настройка категориальных и численных параметров. Важной областью применения предложенного метода является настройка метаэвристических алгоритмов оптимизации. Его эффективность продемонстрирована на примере настройки алгоритма роевого интеллекта.

Примечание

Данная работа не была финансово поддержана ни одной академической или иной организацией. Если изложенный материал был вам полезен, просьба поддержать автора: <http://inversed.ru/Donate.htm>

Литература

1. *Sean Luke*, Essentials of Metaheuristics. [Электронный ресурс] – Режим доступа: <http://cs.gmu.edu/~sean/book/metaheuristics/>
2. *Adenso-Diaz, B., Laguna, M.* Fine-tuning of algorithms using fractional experimental design and local search // Operations Research – 2006 – 54(1) – pp. 99–114.
3. *Hutter, F., Hoos, H. H., & Stutzle, T.* Automatic algorithm configuration based on local search. Proceedings of the Twenty-second National Conference on Artificial Intelligence (AAAI'07) (Eds. Howe, A. & Holte, R. C.) – pp. 1152–1157. AAAI Press / The MIT Press, Menlo Park, CA, USA, 2007.
4. *Telford, J. K.* A Brief Introduction to Design of Experiments // Johns Hopkins APL Technical Digest – 2007 – 27 – pp. 224–232.
5. *Czitrom, V.* (1999). One-Factor-at-a-Time versus Designed Experiments // The American Statistician – 1999 – 53 – pp. 126–131.
6. *Frey, D. D., Engelhardt, F., and Greitzer, E. M.* (). A Role for One Factor at a Time Experimentation in Parameter Design // Research in Engineering Design – 2003 – 14 – pp. 65–74.
7. TSPLIB. [Электронный ресурс] – Режим доступа: <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>
8. *Christian Nilsson.* Heuristics for the Traveling Salesman Problem // Tech. Report, Linköping University, Sweden, 2003.
www.ida.liu.se/~TDDB19/reports_2003/htsp.pdf