Настройка параметров эвристического метода Кларка-Райта с помощью генетического алгоритма

Maria Battarra, Bruce Golden, Daniele Vigo
Автор перевода: Александрова О.А.
Источник: http://or.ingce.unibo.it/ricerca/technical-reports-or-ingce/papers/gacw-sito.pdf.

ВВЕДЕНИЕ

Почти все эвристические процедуры оптимизации требуют установки некоторых параметров, таких как веса, используемые в формулах оценки и количество выполняемых итераций. Определение соответствующих значений для таких параметров, то есть, которые предоставляют хорошую производительность для всех соответствующих решаемых примеров, является серьезной проблемой в проектировании и тестировании алгоритмов, которые могут требовать интенсивных вычислительных требований. Поэтому, исследование относительно быстрых и эффективных подходов для настройки параметра имеет значительную важность, как с теоретической, так и с практической точки зрения.

Использование генетических алгоритмов, чтобы решить задачу настройки параметров метода Кларка-Райта, было сначала предложено Голденом и др. *в* 1998, которые использовали двухэтапную процедуру, чтобы определить значения параметра лагранжевых релаксаций для задачи маршрутизации транспортного средства (ЗМТ). Был дан маленький набор примеров представительных случаев задачи, названных анализируемым набором в последствии. На первом шаге процедуры, генетический алгоритм используется, чтобы определить, для каждого отдельного примера, хороший вектор параметров. А на втором шаге, другой генетический алгоритм определяет ряд весов, чтобы объединить различные векторы параметра в один полный вектор. Вычислительное тестирование, по большему набору эталонных случаев из литературы, показало, что этот подход произвел более устойчивые настройки параметра, чем традиционные методы экспериментального расчета. Та же самая методика была применена Пеппером и другими в 2002 году к фазе установления параметров эвристики. Она была основана на отжиге для задачи коммивояжера (ЗК).

Позже Чандран *в* 2003 году ввел более простую одноэтапную процедуру, в которой генетический алгоритм производит параметры, рассматривая анализируемый набор как единое целое. Этот подход был применен с очень хорошими результатами к варианту так называемого алгоритма демона для ЗК.

В этой статье мы далее исследуем этот подход, определяя одноэтапную процедуру, базируемую на генетическом алгоритме для параметрической версии, предложенной Алтинелом и Онканом в 2005 г. известного эвристического метода Кларка-Райта для ЗМТ. В данном подходе выполняется расширение эвристического метода Кларка и Райта несколькими тысячами различных векторов параметра. Алгоритм получил значительное усовершенствование в отношении качества решения относительно исходного эвристического метода Кларка-Райта. Процедура установки параметров, основанная на генетике, экспериментально оказалась способной к получению результатов сопоставимого качества с более коротким вычислительным временем.

1 ЗАДАЧА МАРШРУТИЗАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ И ПАРАМЕТРИЧЕСКИЙ ЭВРИСТИЧЕСКИЙ МЕТОД АЛТИНЕЛА И ОНКАНА

Задача маршрутизации транспортного средства - является важной и трудной комбинаторной задачей оптимизации, которая требует определения оптимального набора маршрутов, используемых парком транспортных средств для обслуживания ряда клиентов, принимая во внимание различные эксплуатационные ограничения. Наиболее изученный вариант этой проблемы - так называемая ЗМТ с учетом грузоподъемности (ЗМТГ), в которой все транспортные средства идентичны, и рассматривают только ограничения транспортного средства по грузоподъемности. Пусть $V = \{0, \ldots, n\}$ будет набором узлов полного ненаправленного графа $G= (V, E)$. Узел 0 представляет склад, где имеются транспортные средства, каждый с грузоподъемностью D, принимая во внимание, что остающиеся узлы представляют n клиентов, каждый связан с неотрицательным спросом di, i =1, …, n. Кроме того, для каждой дуги (i, j) Є E, пусть cij обозначает стоимость этого переезда. ЗМТГ сводится к определению ряда маршрутов с минимальной общей стоимостью таким образом, что каждый клиент посещается единоразово единственным транспортным средством, которое начинает и заканчивает свой маршрут на складе, и полный спрос не превышает грузоподъемность транспортного средства.

Для ЗМТГ был предложенный известный эвристический алгоритм Кларка-Райта. Этот известный эвристический метод использует понятие *выигрышей,* чтобы оценить операции слияния между маршрутами. Выигрыши - мера сокращения стоимости, полученная комбинированием двух маленьких маршрутов в один больший маршрут. Даны двое клиентов i и j, связанная экономия определена следующим образом:

$$\mathrm{S_{ij} = C_{i0} + C_{0j} - C_{ij}} \tag{1}$$

Алгоритм начинается с решения, в котором каждый клиент обслуживается один на маршруте. Затем, для всех пар клиента вычисляется выигрыш, и выигрышный список сортируется от наибольшего до наименьшего. При каждом повторении алгоритма рассматриваются следующие выигрыши, и если два связанных клиента, которые принадлежат двум различным маршрутам, могут быть допустимо слиты в новый маршрут, тогда маршруты сливаются. Алгоритм Кларка и Райта очень быстр и относительно легко осуществим, однако, учитывая его жадный характер, полученные решения имеют часто недостаточное качество относительно более сложных подходов. В частности нужно отметить, что алгоритм Кларка и Райта не позволяет контролировать итоговое число полученных маршрутов, так как вероятность слияний маршрута может решительно уменьшиться после первых нескольких итераций в сильно ограниченных задачах. Несколько попыток были сделаны в литературе для представления измененных зависимых от параметра выигрышных формул, которые приводят к лучшим результатам в целом. Гаскелл в 1967г. ввел параметр λ (названный параметром *формы маршрута*), который управляет относительной важностью прямой дуги между этими двумя клиентами в вычислении выигрыша. Паесенс в 1988г. включил в выражение выигрыша новый термин, вместе с весовым параметром μ, который учитывает "асимметрию" по расстоянию от склада до каждого из двух "слитых" клиентов. Результирующая формула выигрыша такая:

$$S_{ij} = C_{i0} + C_{0j} - \lambda C_{ij} + \mu \left| C_{0i} - C_{j0} \right| \tag{2}$$

Алтинел и Онкан далее изменили параметрическое выражение выигрыша при попытке рассмотреть двойственную природу ЗМТГ. В частности ЗМТГ комбинирует структуру многомерной ЗМТ (m-ЗМТ) и задачи упаковки рюкзака (ЗР). Ранее предложенная формула выигрыша не рассматривала выгоду полного использования грузоподъемности транспортного средства, следовательно Алтинел и Онкан представили третий весовой термин в формуле выигрыша, чтобы сосредоточиться на спросах клиента, согласно идее "*больший комбинированный маршрут является лучшим*":

$$S_{ij} = C_{i0} + C_{0j} - \lambda C_{ij} + \mu \left| C_{0i} - C_{j0} \right| + \nu (d_i + d_j) / \overline{d} \tag{3}$$

где $\overline{d}$ - средний спрос клиента.

Новое выражение требует настройки трех независимых параметров (λ, μ, ν). До этого времени Алтинел и Онкан использовали простой исчисляющий подход. Они выполняли алгоритм со значениями параметров, изменяющихся в диапазонах [0.1, 2] для параметра λ и [0, 2] для параметров μ и ν, используя размер шага, равный 0.1. Таким образом, были получены 8820 различных векторов параметра. Итоговый вычислительный алгоритм оказался лучше, чем оригинальный эвристический Кларка-Райта, который соответствует вектору параметра (1,0,0), но ясно требует намного большего вычислительного времени, которое чрезмерно обременительно, когда число клиентов является большим.

2 ПРОЦЕДУРА НАСТРОЙКИ ПАРАМЕТРОВ

В этом разделе мы описываем одноэтапную процедуру, устанавливающую параметр, которую мы предлагаем для определения хороших значений параметров, которые будут использоваться с формулой выигрыша. До этого мы используем генетический алгоритм, который исследует область параметра, чтобы определить маленькое подмножество векторов Q, которые способны привести к лучшим средним результатам, когда применяется к случаям анализируемого набора. Тогда, для каждого случая, полный алгоритм, применяет параметрический выигрышный алгоритм, используя только векторы параметра, принадлежащие множеству Q и полученные с ограниченным локальным шагом поиска. Так уже сделанный в предыдущих работах генетический алгоритм используется для процесса настройки, так как этот тип алгоритма является особенно подходящим для решения неограниченных задач оптимизации, таких как задача настройки, которую мы рассматриваем.

Генетические алгоритмы принадлежат классу метаэвристики и являются адаптивными процедурами, основанными на принципах естественного отбора и выживании сильнейших. Цель генетического алгоритма - оптимизация фитнесс-функции, которая выражает производительность определенного решения. Генетический алгоритм начинается с начальной популяции решений (обычно сгенерированной случайным образом). Во время итераций или поколений, исследуется область решения. Как с эволюцией и естественным отбором, особи с самыми перспективными характеристиками и лучшими фитнесс-значениями, более вероятно, участвуют в репродукции, и, более вероятно, выживут. Каждая итерация генетического алгоритма состоит из следующих шагов:

1) Кроссинговер. Самые перспективные особи в популяции с точки зрения значения фитнесс-функции, выбираются как родители для воспроизводства. Потомство произведено, и оно отражает генетическое наследие их родителей.

2) Мутация. Генетические характеристики каждого потомства изменены после вероятностных правил.

3) Выбор. Каждое потомство, после мутации, оценивается с точки зрения ее пригодности. Если это значение высокое, потомство вводится в популяцию (и другая менее пригодная особь, как правило, удаляется), иначе потомство прерывается.

Шаги 1-3 повторяются, пока критерий останова не удовлетворен.

Как ранее было упомянуто, одношаговая процедура установки параметра применена к подмножеству случаев задачи, названных анализируемым набором. Примеры, выбранные в анализируемом наборе, должны быть представительными для примеров ЗМТГ, которые будут решены, но их число должно быть сохранено таким маленьким, как этого требует умеренное вычислительное время для оценки фитнесс-функции для каждого набора параметра в данном поколении генетического алгоритма.

В нашем устанавливающем параметр генетическом алгоритме особи популяции соответствуют различным векторам этих трех параметров (λ, μ, ν) в формуле выигрыша (3).

Начальная популяция содержит 25 особей, полученных случайным выбором каждого параметра в пределах диапазонов, используемых Алтинелом и Онканом, то есть, [0.1, 2] для параметра λ и [0, 2] для параметров μ и ν. Для данного параметрического вектора *w*, фитнесс-функция измеряет по всем примерам анализируемого набора, среднее квадратичное относительного дополнительного отклонения решений *D* (*w*, *i*) полученных с параметрическим вектором *w*, относительно самого известного решения примера *i* обозначенного *B (i*). Отметим, что отношение *D* (*w*, *i*)/*B* (*i*) не меньше чем 1, поэтому, 1 вычитается из отношения, чтобы дать лишнее отклонение. В общем виде фитнесс-функция, которую необходимо минимизировать может быть представлена следующим выражением:

$$F(\omega) = 100 \cdot \sqrt{\frac{1}{|P|} \cdot \sum_{i=1}^{|P|} ((D(\omega, i) / B(i)) - 1)^2} \quad (4)$$

Генетический алгоритм итерационно развивает особей популяции, чтобы получить особей следующего поколения посредством следующих операторов:

1) Кроссинговер. Находим лучшую особь в популяции, называемую королевой - матерью. Потомство генерируется как результат скрещивания королевы-матери и каждой другой особи в популяции. Эти другие особи - отцы. В данном потомстве значение каждого параметра случайно производится из равномерного распределения в диапазоне, определенном соответствующими значениями параметра родителей.

2) Мутация. Для каждого потомства значение каждого параметра мутирует с вероятностью *r*, случайно производя новое значение согласно равномерному распределению в исходном диапазоне.

3) Отбор. Лучшие 25 особей с точки зрения значения фитнесс-функции промежуточной популяции, составленной из текущего набора 25 родителей плюс 24 потомка, произведенных двумя предыдущими операторами, отбираются как текущая популяция для следующей итерации алгоритма.

Итерационный процесс останавливается, как только среднее значение фитнесс-функции особей в популяции не уменьшается в последних h поколениях.

Для определения множества *Q* параметрического вектора, чтобы использовать в нашем полном алгоритме, мы переходим к следующему. Во время генетического алгоритма мы храним все различные векторы параметра, которые были произведены. В конце выполнения мы выбираем вектор с наименьшей фитнесс-фунцией. Мы выбираем в большинстве |*Q* |-1 дополнительных векторов те, фитнесс-значения которых не более, чем на 20 % выше, чем у лучшего, и это максимизирует "расстояние" от лучшего вектора. Как расстояние между двумя векторами, мы использовали сумму абсолютных разностей соответствующих значений параметра.

В нашем эвристическом методе мы запускаем алгоритм Кларка-Райта с формулой выигрыша с небольшим количеством векторов параметра, включая те в набор Q и некоторых полученных, выполняя ограниченный локальный поиск вокруг них. Более точно, локальный поиск рассматривает маленькую окрестность, включающую 14 точек, соответствующих 8 вершинам и 6 центрам граней куба, длина ребра которого 0.2 и чей центр в точке, координаты которой – значения параметра текущего вектора. Параметрический метод Кларка-Райта запускается для каждого вектора в окрестности, и лучший, еще не использованный, отбирается как новый текущий вектор. Для каждого вектора множества *Q*, локальный поиск повторяется три раза, получая 41 различный вектор, а именно, стартовый, плюс 14 векторов первой окрестности и 13 векторов каждой последующей окрестности. Так как |*Q* | = 5, наш полный алгоритм требует только 205 выполнений параметрического алгоритма Кларка-Райта. Это значение более чем в 40 раз меньше, чем число выполнений, требуемых в соответствии с исчисляющей процедурой Алтинела и Онкана.

3 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

В этом разделе мы анализируем результаты, полученные во время обширного вычислительного анализа эвристического метода, описанного в этой статье. Главная цель этого анализа состоит в том, чтобы проверить, в состоянии ли одношаговый подход, устанавливающий параметры, который производит маленькое подмножество векторов параметра, используемых алгоритмом Кларка-Райта, найти векторы, которые производят со значительно меньшим вычислительным временем решения, имеющие сопоставимое качество к тем, которые могут быть получены исчисляющим подходом, устанавливающим параметр, предложенным Алтинелом и Онканом.

В наших экспериментах мы использовали те же самые тестовые примеры ЗМТГ рассмотренные как эталонный набор. Все примеры могут быть загружены на сайте http://www.branchandcut.org. Как анализируемый набор, мы выбрали (из эталонного набора) пять примеров, отличающихся в отношении числа клиентов и пространственного местоположения склада. Анализируемый набор составлен из примеров, приведенных в таблице 1:

Таблица 1 Примеры анализируемого набора

Name	*n*+1	*D*	Test set	Depot Location	Best Known Solution
P-n16-k8	16	35	(Augerat *et al*, 1995), set P	bottom-left	450
A-n53-k7	53	100	(Augerat *et al*, 1995), set A	centre-left	1010
B-n78-k10	78	100	(Augerat *et al*, 1995), set B	bottom-centre	1221
E-n101-k14	101	112	(Christofides and Elion, 1969)	centre	1067
CMT199	199	200	(Christofides *et al*, 1979)	centre	1291.45

Генетический алгоритм, устанавливающий параметр, был реализован на Matlab и запущен на AMD Athlon TM XP Processor 3000+, 797 MHz и 256 Mb ОП. Используя этот алгоритм, с $|Q| = 5$ мы определили, в течение приблизительно 12 минут вычислительного времени, пять векторов параметра, приведенных в таблице 2.

Таблица 2 $|Q|$ векторов параметра, произведенных генетическим алгоритмом

Vector	Fitness value	Distance from vector 1	λ	μ	ν
1	5.0564	-	1.5578	0.6920	0.8190
2	5.3053	1.4457	0.8830	0.6948	1.5871
3	5.3704	1.3165	0.7335	0.6657	1.2849
4	5.9371	1.1883	1.4891	0.6404	1.8870
5	5.4035	0.9719	1.6442	0.7251	1.6714

Результаты алгоритма, когда применяется весь набор 87 эталонных примеров получены в таблице 3 и сравнены с таковыми в исчисляющем алгоритме Алтинела и Онкана. Более точно, для каждого из пяти классов в эталонном наборе и для каждого алгоритма, таблица сообщает о среднем вычислительном времени в секундах, и среднем процентном усовершенствовании решений, полученных алгоритмом, над полученными исходным алгоритмом Кларка-Райта.

Таблица 3 Сравнение метода, устанавливающего параметр с и без Локального Поиска относительно подхода Алтинела и Онкана

Test set	# of instances	Altınel and Öncan		BGV		BGV with Local Search	
		Average Time (sec)	% solution improvement over CW	Average Time (sec)	% solution improvement over CW	Average Time (sec)	% solution improvement over CW
(Augerat *et al*, 1995), set A	27	35.76	2.44	0.02	1.13	0.97	1.84
(Augerat *et al*, 1995), set B	23	39.27	2.10	0.01	0.88	1.06	1.80
(Augerat *et al*, 1995), set P	22	33.20	4.35	0.02	2.52	0.88	1.96
(Christofides and Elion, 1969)	8	50.12	3.05	0.03	1.84	1.25	1.96
(Christofides *et al*, 1979)	7	179.26	3.06	0.10	1.07	4.37	1.96
Total	87	48.91	2.94	0.02	1.47	1.27	2.22

Анализируя таблицу 3, можно заметить, что предложенный алгоритм без локального шагового поиска является быстрее более чем на три порядка, чем алгоритм Алтинела и Онкана и все еще в состоянии получить приблизительно половину усовершенствования относительно оригинального подхода Кларка-Райта. Действительно, это улучшает Кларка-Райта примерно на 1.5 %, в то время как Алтинел и Онкан улучшают меньше, чем на 3 %. Вводя ограниченный локальный шаговый поиск, вычислительные времена все еще приблизительно в 40 раз меньше, и усовершенствование - фактически то же самое как у Алтинела и Онкана. В обоих случаях, мы не включали вычислительное время, требуемое генетической процедурой настройки параметров, которая запускается один раз для всех примеров. С другой стороны, даже если мы включаем время на установку параметров, равное 720 секундам в нашей реализации на Matlab, полное время, требуемое нашим подходом для целого набора 87 примеров, все еще более чем в пять раз быстрее, чем у Алтинела и Онкана.

Мы должны заметить, что вычислительное время процедуры настройки параметров, может быть значительно сокращено при использовании эффективной реализации на языке C. Кроме того, учитывая значительное разнообразие примеров, которые использовались в вычислительном тестировании, вероятно, что набор параметров, определенных во время настройки параметра, может использоваться с другими примерами с сопоставимыми результатами, не требуя определенных настраивающих шагов.

ЛИТЕРАТУРА

1. Altınel I K, Öncan T (2003). A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. FBE-IE-02/2003-02. Institute for Graduate Studies in Science and Engineering, Bogaziçi University, Bebek, Istanbul, Turkiye.

2. Altınel I K, Öncan T (2005). A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. J Opl Res Soc 56: 954-961.

3. Augerat P, Belenguer J M, Benavent E, Corberan A, Naddef D, Rinaldi G (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical report RR 949-M, University Joseph Fourier, Grenoble, France.

4. Chandran B, Golden B, Wasil E (2003). A Computational Study of Three Demon Algorithm Variants for Solving the Travelling Salesman Problem. In: Barghava HK and Ye N (eds).

Computational Modelling and Problem Solving in the Networked World: Interfaces in Computer Science and Operations Research. Operations Research/Computer Science Interfaces Series, Kluwer Academic Publisher: Boston, MA, pp 155-175.

5. Christofides N, Eilon S (1969). An algorithm for the vehicle routing dispatching problem. Opl Res Quart 20: 309-318.

6. Christofides N, Mingozzi A, Toth P(1979). The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P, Sandi C (eds). In: Combinatorial Optimization. Wiley, Chichester, pp 315-338.

7. Clarke G, Wright J (1964). Scheduling of vehicles from a central depot to a number of delivery points. Opns Res 12 (4): 568-581.

8. Coy S (1998). Fine-tuned Learning: A New Approach to Improving the Performance of Local Search Heuristics. Ph.D.Dissertation, Univ.of Md.,College Park.

9. Gaskell T J (1967). Bases for vehicle fleet scheduling. Opl Res Quart 18: 281-295.

10. Golden B, Pepper J, Vossen T (1998). Using genetic algorithms for setting parameter values in heuristic search. In: Intelligent Engineering System through Artificial Neural Networks, ASME Press, New York, 8: 239-245.

11. Holland H (1992). Adaptation in Natural and Artificial Systems. MIT Press: Ann Arbor, MI.

12. Michalewicz Z (1996). Genetic Algorithms + Data Structures = Evolution Programs, Third, revised and Extended Edition. Springer-Verlag Berlin Heidelberg New York Publishing: Berlin.

13.Paessens H (1988). The savings algorithm for the vehicle routing problem. Eur J Opl Res 34: 336-344.

12. Pepper J, Golden B, Wasil E (2002). Solving the travel salesman problem with annealing-based heuristics: a Computational Study. IEEE Transactions on Systems, Man and Cybernetics A 32(1): 72-77.

13. Toth P, Vigo D (2001). The vehicle routing problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM Publishing: Philadelphia, PA.

14. Yellow P (1970). A computational modification to the savings method of vehicle scheduling. Ops Res Quart 21: 281-283.