УДК 519.8

Методы оценки эффективности алгоритмов решения многокритериальных задач

© Е.А. Лазарев[1]

Аннотация. В работе предлагаются несколько методов оценки эффективности алгоритмов решения многокритериальных задач, приводится их сравнение и критический анализ.
Ключевые слова: Многокритериальная оптимизация, множество Парето, количественная оценка эффективности алгоритма.

1. Введение

При рассмотрении многокритериальных задач оптимизации знание множества Парето помогает лицу принимающему решение выбрать наилучшее компромиссное решение [1].

Однако, на практике, нахождение всего множества Парето может быть очень вычислительно трудоемко и, зачастую, просто невозможно, поэтому применение точных методов нерационально. Одним из возможных решений данной проблемы является использование класса эволюционных алгоритмов (ЭА). Они, зачастую, не гарантируют нахождения точного фронта Парето, но позволяют найти достаточно хорошее его приближение, т.е. множества решений, которые достаточно близки к парето-множеству. Существует большое количество различных эволюционных алгоритмов, и нас, естественно, интересует способ выбора наилучшего метода для решения поставленной задачи оптимизации.

Понятие эффективности алгоритма обычно включает в себя две компоненты: качество выдаваемого результата и вычислительные ресурсы, требуемые для его получения. Во втором случае, общей практикой является использование фиксированного числа операций вычисления приспособленности или константного времени выполнения в целом, поэтому в этом смысле нет разницы между однокритериальной и многокритериальной оптимизацией. Однако, при оценке качества выдаваемого результата существует серьезная разница. В случае однокритериальной оптимизации качество решения может быть определено с помощью целевой функции – чем больше (меньше) значение, тем лучше решение. Однако, при рассмотрении многокритериальных задач неясно, что означает качество – близость к фронту Парето, покрытие широкого спектра разнородных решений или оно характеризуется какими-то другими признаками?

2. Постановка задачи

Пусть рассматривается задача оптимизации с n критериями, которые, без ограничения общности, должны быть минимизированы. Обозначим множество допустимых решений задачи через X в смысле значения целевых функций. Каждый элемент $x \in X$ называется *целевым вектором* и имеет вид $x = (x^1, x^2, \ldots, x^n)$, где $x^1, x^2, \ldots, x^n$ – значения целевых функций. В дальнейшем термины решение и целевой вектор используются как взаимозаменяемые.

[1] Аспирант кафедры вычислительные системы и технологии, Нижегородский государственный технический университет имени Р.Е. Алексеева, г. Нижний Новгород; elazarev.nnov@gmail.com.

Рассматривается наиболее общий случай, когда все целевые функции имеют равную важность, т.е. нет никакой дополнительной информации о решаемой задаче. Единственное предположение заключается в том, что решение x_1 предпочтительнее решения x_2 если оно не хуже чем x_2 по всем критериям и лучше по, как минимум, одному. Данное отношение известно как *доминирование по Парето* и, в этом случае, говорят что x_1 доминирует x_2 и обозначают $x_1 \succ x_2$. Отношение доминирования задает частичное упорядочивание на пространстве поиска, таким образом, оптимальным может быть определено решение, которое не доминируется ни одним другим решением. Однако, могут существовать,несколько таких решений, которые называются *оптимальными по Парето*, так как два целевых вектора могут быть несравнимы друг с другом: один превосходит другой по некоторым критериям и уступает по другим. Например, несравнимы два целевых вектора, описываемые парами величин $(10, 8)$ и $(7, 11)$, так как $10 > 7$, а $8 < 11$.

Большая часть работ в области эволюционной многокритериальной оптимизации посвяшена поиску оптимальных по Парето решений или, если это невозможно или крайне сложно с вычислительной точки зрения, нахождению хорошего их приближенного множества. Принимая во внимание данный факт, дадим определение множеству недоминируемых решений, находимых эволюционным алгоритмом:

Определение 2.1. *Множество решений $A \subseteq X$ назовем субоптимальным, если каждый элемент из A не доминирует и не равен ни одному другому элементу из A. Множество всех суботимальных множеств решений обозначим Ω.*

Понятно, что решения, найденные эволюционным алгоритмом, доминируемые каким-то другим решением, не представляют интереса и, поэтому, могут быть отброшены. Так же отметим, что данное выше определение, не включает в себя никакого представления о качестве полученного множества решений. Однако, нас интересует не любое субоптимальное решение, а *хорошее* субоптимальное решение. В идеальном случае задача состоит в нахождении *фронта Парето*, который состоит из всех парето-оптимальных решений. Для того, чтобы определить качество полученного субоптимального решения рассмотрим несколько различных методов.

3. Методы оценки

В математическом анализе для вычисления расстояния между двумя множествами A и B обычно используется Хаусдорфово расстояние, которое вычисляется следующим образом. Для каждого элемента $a \in A$ находится ближайший элемент $b \in B$. Среди найденных расстояний выбирается наибольшее. Обозначим его A_m. Затем, по аналогии, для каждого $b \in B$ находится ближайший элемент $a \in A$ и снова выбирается наибольшее из полученных расстояний. Обозначим его за B_m. Тогда Хаусдорфово расстояние между A и B есть $max(A_m, B_m)$.

Однако, использование Хаусдорфовой меры для оценки качества полученного субоптимального решения не является эффективным. Такая мера позволит показать в каких «границах» находится субоптимальное решение от множества Парето. Например, пусть фронт Парето X имеет следующий вид $X = \{(1, 4), (2, 8), (7, 9)\}$, а два субоптимальных решения следующие $X_1 = \{(1, 4), (4, 7), (7, 9)\}$, $X_1 = \{(1, 4), (4, 7), (8, 9)\}$. Хаусдорфово расстояние от X до X_1 и X_2 одинаково, однако решение X_1 предпочтительнее ввиду того, что в нем только одно решение $(4, 7)$ не является парето-оптимальным, а в X_2 сразу два: $(4, 7)$ и $(8, 9)$.

Таким образом, необходимо разработать метод оценки субоптимального решения, который бы адекватно показывал точность полученного решения.

Прежде чем перейти к описанию методов оценки введем несколько определений, которые будут использоваться в дальнейшем.

Обозначим через K фронт Парето для рассматриваемой задачи оптимизации, состоящий из $p \in \mathbb{N}$ элементов, а через $K' \subseteq \Omega$ – субоптимальное множество решений, состоящее из $q \in \mathbb{N}$ элементов, найденное эволюционным алгоритмом при решении той же задачи.

Определение 3.1. *Отклонение $\Delta(K, K')$ субоптимального множества решений K' от фронта Парето K есть функция $\Delta : \mathbb{R}^{p\times n} \times \mathbb{R}^{q\times n} \to \mathbb{R}$.*

Определение 3.2. *Решение $x \in K'$ назовем хорошим, если $x \in K$.*

Определение 3.3. *Решение $x \in K'$ назовем плохим, если $x \notin K$.*

Расстоянием $\rho(x_1, x_2)$ между решениями x_1 и x_2 назовем расстояние по метрике Чебышева:

$$\rho(x_1, x_2) = \max_{i=1\ldots n}(|x_1^i - x_2^i|). \tag{3.1}$$

Количество плохих решений Q_b вычисляется по формуле:

$$Q_b = |\{x : x \in K', x \notin K\}| \tag{3.2}$$

Количество хороших решений Q_g вычисляется по формуле:

$$Q_g = |\{x : x \in K', x \in K'\}| \tag{3.3}$$

3.1. Метод подсчета решений

Суть этого метода состоит в подсчете количества Q_m парето-оптимальных решений, которые не были найдены эволюционным алгоритмом и числа Q_b полученных плохих решений. Затем, наибольшая из найденных величин делится на количество элементов в K.

Величина Q_m вычисляется по формуле:

$$Q_m = |\{x : x \in K, x \notin K'\}| \tag{3.4}$$

Тогда отклонение Δ_1 вычисляется по следующей формуле:

$$\Delta_1(K, K') = \frac{\max(Q_m, Q_b)}{|K|} \tag{3.5}$$

Данный метод позволяет достаточно грубо оценить качество полученной субоптимальной совокупности по количеству плохих и ненайденных оптимальных решений. Недостатком этого подхода является то, что при вычислении отклонения не учитывается насколько сильно отстоит субоптимальная оценка от оценки соответствующеего решения из совокупности Парето.

Например, пусть есть два субоптимальных множества $X_1 = \{(10, 15)\}$ и $X_2 = \{(5, 6)\}$, состоящих из одного решения. Расстояние между оценками и фронтом Парето $X = \{(5, 5)\}$ для первого равна $\max(|10 - 5|, |15 - 5|) = 10$, а для второго равна $\max(|5 - 5|, |6 - 5|) = 1$. Таким образом, несмотря на то, что оба решения вносят одинаковый вклад в величину отклонения, на практике выбор X_2 является предпочтительнее.

3.2. Метод усреднения отклонения от точного решения

Данный метод предложен в работе [2]. Для каждой оценки из множества Парето находится ближайшая к ней оценка из множества субоптимальных решений, а затем производится усреднение по всем оценкам из точной совокупности. Полученная величина нормируется на среднее значение расстояния парето-оптимальных решений до нуля. Таким образом, отклонение вычисляется по следующей формуле:

$$\Delta_2(K,K') = \frac{\underset{x\in K}{avg}\left(\min\limits_{x'\in K'}\rho(x,x')\right)}{\underset{x\in K}{avg}\left(\rho(x,x_0)\right))} \tag{3.6}$$

где $x_0=(0,0,\ldots,0)$, а avg – оператор нахождения среднего значения.

Однако этот метод не лишен недостатков. Такой способ оценки отклонения дает понятие о приближении всей совокупности парето-оптимальных решений в целом, в среднем. Однако при таком подходе ситуации нахождения оценки, «неплохо» приближающей сразу несколько оценок совокупности Парето, иногда оказываются выигрышней ситуации нахождения какой-то одной точной оценки из их числа. В работе [2] приводится пример, иллюстрирующий это.

Для учета подобных ситуаций в работе [2] предлагается внести изменения в алгоритм оценки приближения оптимального множества решений. Если для некоторой оценки из парето-оптимальной совокупности расстояние для ближайшей оценки из субоптимальной совокупности превышает некоторую заданную величину ε, то ее вклад при нахождении среднего отклонения не учитывается, но при этом увеличивается число оценок, потерянных рассматриваемым субоптимальным алгоритмом. При таком подходе при сравнении алгоритмов предпочтительнее оказывается тот, который потерял наименьшее количество оценок из фронта Парето, и при этом, обеспечил наименьшее среднее относительное отклонение для тех оценок, которые он приблизил.

Однако при таком подходе встает вопрос о том, каким образом выбирать константу ε? И если в алгоритме считается количество потерянных решений, то почему бы просто не считать отклонение по формулам (3.5) и (3.6) и переходить к рассмотрению бикритериальной задачи?

Так же стоит отметить, что нормирование отклонения на среднее значение расстояния парето-оптимальных решений до нуля не является оптимальным. Для каждого решения из субоптимального множества стоит производить нормировку на величину расстояния от *ближайшего* элемента из множества Парето до нуля. Таким образом вычисляется на сколько сильно отстоит каждое конкретное субоптимальное решение от ближайшего эталонного. Тогда формула для вычисления отклонения примет вид:

$$\Delta_3(K,K') = \underset{x\in K}{avg}\frac{\min\limits_{x'\in K'}\rho(x,x')}{\rho(x'',x_0)} \tag{3.7}$$

где $x''\in K$ – решение ближайшее к x.

Еще одним недостатком данного метода является то, что не учитываются решения, которые были найдены эволюционным алгоритмом, но при этом не входят в фронт Парето. Таким образом, субоптимальное решение может содержать большое количество «лишних» решений. Для учета этого случая рекомендуется совместное применение формул (3.7) и (3.5), что позволит избежать данной проблемы.

4. Заключение

Рассмотренные выше методы оценки были использованы при решении бикритериальной задачи оптимизации сети передачи данных [3] при анализе двух генетических алгоритмов [4] и алгоритма имитации отжига, решающих поставленную задачу. Совместное использование формул (3.5) и (3.7) позволяет не только оценить насколько сильно, в среднем, отклонение субоптимального решения, но и определить как много решений было потеряно или не является парето-оптимальными.

Стоит отметить, что существуют другие способы оценки качества полученного решения, например, метод, использующий метрику наподобие χ^2, предложен в работе [5], метод расчета гиперобъема, предложенный в работе [6], метод нахождения протяженности фронта Парето [7]. Однако, данные методы либо достаточно сложны в реализации, либо требуют значительных вычислительных ресурсов, либо уступают в точности и, при этом, не обладают серьезным преимуществом по сравнению с описанными выше методами.

Список литературы

1. В. В. Подиновский., *Парето-оптимальные решения многокритериальных задач*, Физматлит, М., 2007.

2. Н. А. Дуничкина, Ю. С. Федосенко., “Об оценке приближенных решений бикритериальных задач дискретного программирования”, *Материалы XV международной научно-технической конференции Информационные системы и технологии*, 2009, 300—302.

3. Е. А. Лазарев, Д. Е. Шапошников, П. В. Мисевич., “Бикритериальная модель сети передачи данных”, *Системы управления и информационные технологии*, 2011, № 3.2(45), 255-–258.

4. Е. А. Лазарев, Д. Е. Шапошников, П. В. Мисевич., “Генетические алгоритмы оптимизации сети передачи данных”, *Системы управления и информационные технологии*, 2011, № 4(46), 59—63.

5. N. Srinivas, K. Deb., “Muiltiobjective optimization using nondominated sorting in genetic algorithms”, *Evol. Comput.*, **2**:3 (1994), 221–248.

6. E. Zitzler, L. Thiele., “Multiobjective optimization using evolutionary algorithms – a comparative case study”, *Fifth International Conference on Parallel Problem Solving from Nature*, 1998, 292–301.

7. J. Wu, S. Azarm., “Metrics for quality assessment of a multiobjective design optimization solution set”, *Journal of Mechanical Design*, 2001, 18—25.

Methods of evaluation of performance of algorithms solving multicriterion optimization problems

© E.A. Lazarev[2]

Abstract. Several methods of evaluation of performance of algorithms solving multicriterion optimization problems are proposed in the work, they are compared and analyzed.
Key Words: multicriterion optimization, Pareto set, quantative comparison of algorithm performance.

[2] Graduate student of the department of computer systems and technologies, Nizhny Novgorod State Technical University after R.E. Alekseev, Nizhny Novgorod; elazarev.nnov@gmail.com.