# О КЛАССИФИКАЦИИ ПРИБЛИЖЕННЫХ МЕТОДОВ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

**Сергей Сиренко**

***Аннотация:*** *В работе предлагается классификация приближенных методов комбинаторной оптимизации, которая обобщает и дополняет существующие подходы.*

***Ключевые слова:*** *комбинаторная оптимизация, классификация, приближенные методы.*

***ACM Classification Keywords****: G.1.6 [Numerical Analysis] Optimization – Stochastic programming, G.2.1 [Discrete Mathematics] Combinatorics – Combinatorial algorithms, I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – Heuristic methods.*

## Введение

Развитие методов комбинаторной оптимизации (КО) уже превысило уровень первичного накопления знаний и идет путем обобщения полученных результатов. Это проявляется, в частности, в увеличении внимания к исследованию существующих подходов, анализе их общих и отличных характеристик, а также концептуальных оснований позволяющих достичь повышения эффективности алгоритмов. Описание общих характеристик происходит путем формулирования определенных обобщенных поисковых процедур, конкретизацией отдельных блоков которых обозначаются классы приближенных алгоритмов. Выделение принципиальных отличий может происходить, например, путем классифицирования алгоритмов.

В ряде работ рассматриваются вопросы классификации приближенных алгоритмов КО в целом и отдельных их классов, в частности [Birattari et al., 2001; Blum and Roli, 2003; Laguna, 2002]. В данной работе предлагается обобщение и дополнение вышеупомянутых подходов к классификации. Предлагаемая классификация будет изложена, после короткого описания понятия задачи КО.

## Определение задачи комбинаторной оптимизации

Приведем формальное определение задачи КО [Гуляницкий, 2008]. Пусть заданы $Y = \{1,...,m\}$, $Z$ – дискретное пространство, $\varphi$ – гомоморфизм, $\varphi : Y \to Z$, удовлетворяющий некоторой системе ограничений $\Omega$.

**Определение 1**. Под комбинаторным объектом $\kappa$ будем понимать триаду $\kappa = (\varphi, Z, \Omega)$.

**Определение 2**. Задачей КО, называется задача нахождения такого $x_* \in X$, что

$$x_* = arg \min_{x \in D \subseteq X} f(x),$$

где $X$ – пространство решений задачи, элементами которого являются комбинаторные объекты, $D \subseteq X$ – подпространство допустимых решений, определяемое некоторым предикатом $\Pi$, $f : X \to \mathbb{R}$ – заданная целевая функция задачи.

## Классификация приближенных методов комбинаторной оптимизации

При классифицировании алгоритмов КО прежде всего следует разделять точные и приближенные методы. Первые владеют свойством гарантировано находить оптимальное решение. Однако, почти все практически важные задачи КО являются NP-полными [Пападимитриу и Стайглиц, 1985]: для них не известно ни одного точного алгоритма с полиномиальной сложностью. Кроме того, постановки задач часто бывают с данными, которые имеют определенные погрешности, что окончательно делает значительные вычислительные затраты для нахождения точного решения неоправданными. В противоположность точным методам, современные приближенные алгоритмы КО разрешают получать решения "высокого качества" за приемлемое (с практической точки зрения) время. По этим, а также по другим причинам, все больше внимания в научной литературе уделяется приближенным методам КО. За последние десятилетия предложены десятки различных подходов, обладающие теми или иными свойствами и характеристиками. Для выделения характерных отличных черт существующих приближенных методов КО и предложена данная классификация (см. Таблицу 1). В примерах, приведенных в Таблице 1, ссылки указаны для тех методов которые далее в тексте не упоминаются.

Таблица 1. Классификация приближенных методов комбинаторной оптимизации

| **Классы / Характеристики** | | **Примеры** |
|---|---|---|
| **По принципу принятия решений** | | |
| **Детерминированные** | | *Локальный поиск* |
| **Недетерминированные (стохастические)** | | *Имитационный отжиг* |
| **По структуре** | | |
| **Простой алгоритм** | | *Локальный поиск* |
| **Гибридный алгоритм** | | – |
| **Метаэвристика** | | *Метод муравьиных колоний, меметический алгоритм* |
| **Гибридная метаэвристика** | | – |
| **По типу используемых пространств** | | |
| **Последовательные (конструктивные)** | | *Эвристика "иди к ближайшему"* |
| **Итерационные** | **Траекторные** | *Локальный поиск, G-алгоритм* [Гуляницкий, 2006] |
| | **Популяционные** | *Метод муравьиных колоний, H-метод* [Гуляницкий, 2008] |
| **По количеству используемых систем окрестностей** | | |
| **Одна** | | *Имитационный отжиг* |
| **Много** | | *Поиск в переменных окрестностях* |
| **По типу целевой функции** | | |
| **Статическая** | | *Метод муравьиных колоний, генетический алгоритм* |
| **Динамическая** | | *Управляемый локальный поиск* |
| **По типу траектории (для итерационных методов)** | | |
| **Траекторно-непрерывные** | | *G-алгоритм, табу-поиск* |
| **Траекторно-разрывные** | | GRASP |

| | |
|---|---|
| **По использованию памяти** | |
| **Без использования памяти** | *Локальный поиск, G-алгоритм, стандартный генетический алгоритм* |
| **С кратковременной памятью** | *Простой табу-поиск* |
| **С долговременной памятью** | *Динамический локальный поиск* [Hoos and Stützle, 2005] |
| **С обоими видами памяти** | *Табу-поиск с использованием критериев стремления* |
| **По оценке точности** | |
| **С апостериорной оценкой точности** | *Локальный поиск с известной нижней границей* |
| **С априорной оценкой точности** | *$\varepsilon$-приближенные алгоритмы* |
| **Без оценки точности** | – |
| **Сходимость по решению** | |
| **Сходящиеся с определенной вероятностью** | *Муравьиные алгоритмы класса* $ACO_{bs,\tau_{\min}(t)}$ [Dorigo and Blum, 2005] |
| **Не сходящиеся** | *Стандартный генетический алгоритм* |
| **Без оценки сходимости** | *H-метод* |
| **Сходимость по значению** | |
| **Сходящиеся с определенной вероятностью** | *Муравьиные алгоритмы класса* $ACO_{\tau_{\min}}$ [Dorigo and Blum, 2005], *G-алгоритм* |
| **Не сходящиеся** | *Стандартный генетический алгоритм* |
| **Без оценки сходимости** | *H-метод* |

***По принципу принятия решений.*** В приближенных методах, прежде всего, будем различать два принципа принятия решений – детерминированный и недетерминированный, т.е. отсутствие или наличие псевдослучайных аргументов в функции принятия решения соответственно. На данный момент, подавляющее большинство разработанных и применяемых на практике приближенных методов КО принадлежат именно к недетерминированным, что обусловлено более высокими показателями эффективности таких методов.

***По структуре***. Классифицирование по структуре, хотя и является важным, но, учитывая отсутствие формального определения метаэвристики, не дает возможности четко различить алгоритмы по принадлежности к простым или метаэвристическим. Например, алгоритм имитационного отжига [Kirkpatrik et al., 1983] разные авторы относят как к метаэвристикам, так и считают простым учитывая то, что его отличие от локального поиска является незначительным, даже по сравнении с такой "примитивной" метаэвристической концепцией, как повторяемый локальный поиск [Lourenço et al., 2002]. Метаэвристические методы можно понимать как комбинацию двух техник: общая схема строится на базовом методе, в которую включается та или иная встроенная процедура. Важным аспектом есть то, что встроенная процедура – это в большинстве случаев самостоятельный алгоритм решения той же задачи, что и метаэвристический метод в целом. Под гибридными алгоритмами понимается такое объединение простых алгоритмов, которое еще не порождает метаэвристичний алгоритм. Например, это может быть

простое последовательное или параллельное выполнение. На практике простые алгоритмы используются все реже, поскольку развитие метаэвристических подходов позволяет получать лучшие результаты, а темпы развития вычислительной техники позволяют применять все более сложные алгоритмы к задачам все большей размерности. Классификацию гибридных метаэвристик детально рассмотрено в работе [Raidl, 2006]. По аналогии с последней, можно предложить классифицировать гибридные алгоритмы по степени гибридизации, порядку выполнения и стратегии управления.

***По типу используемых пространств***. Дальше выделим по типу используемых пространств класс последовательных, или же конструктивных, алгоритмов которые состоят в построении решения из отдельных частей по определенным правилам. Такие алгоритмы работают в некотором расширенном пространстве $S \supseteq X$. Во многих случаях это расширенное пространство можно представить в терминах Определения 1 в таком виде:

$$S = X \cup \bigcup_{i=1,m} X_i, X_i = \{x^i = (\varphi^i, Z, \Omega^i) : \varphi^i : \{y_1, ..., y_i\} \to Z\}, i = 1, ..., m;$$

$$\Omega^i \supseteq \Omega^{i-1}, i = 2, ..., m; \{y_1, ..., y_m\} = Y.$$

Простейшим примером является эвристика для задачи коммивояжера, которую часто называют "жадной" или "иди к ближайшему" [Hoos and Stützle, 2005]. На практике конструктивные алгоритмы чаще всего используются либо для задач чрезвычайно большой размерности или задач с затратных вычислением целевой функции, либо для генерирования начальных решений для других методов. Иногда они встроены в состав методов более сложным образом, как, например, в оптимизации муравьиными колониями [Dorigo and Stützle, 2004], конструктивный блок – деятельность искусственных муравьев играет ключевую роль.

Другой класс алгоритмов – итерационные. Такие алгоритмы работают в пространстве полных решений $X$. Итерационные методы естественно разделяют на два класса по размеру множества решений, которыми алгоритм одновременно оперирует: в случае одноэлементного множества алгоритмы называют траекторными (реже, одноточечными), а в случае, когда множество текущих решений содержит больше одного элемента – популяционными.

В работе [Birattari et al., 2001] предлагается также различать методы комбинаторной оптимизации по количеству различных систем окрестностей которые они используют, типу целевой функции и по типу траектории. Рассмотрим эти характеристики более детально.

***По количеству используемых систем окрестностей.*** По количеству используемых окрестностей можно различить алгоритмы, которые используют в процессе поиска одну или несколько различных систем окрестностей. Формально системы окрестностей обычно определяют так [Blum and Roli, 2003].

**Определение 3**. Система окрестностей – это такое отображение $N : X \to 2^{|X|}$, которое каждому решению $x \in X$ ставит в соответствие некоторое множество соседей, которое называется окрестностью решения $x$.

Большинство приближенных алгоритмов КО используют один тип окрестностей. Это такие алгоритмы как имитационный отжиг, табу поиск [Blum and Roli, 2003]. В повторяемом локальном поиске используется по крайней мере две системы окрестностей. В первой осуществляется локальный поиск до достижения локального минимума и после этого осуществляется возмущение найденного локального минимума. Фактически, возмущение может рассматриваться как шаг в другой системе окрестностей. Эти соображения, в частности, были развиты в алгоритме поиска в изменяемых окрестностях [Mladenovic and Hansen, 1997] – в этом подходе окрестности систематически изменяются в пределах заведомо заданного перечня $\{N_1, ..., N_l\}$. Процесс конструирования решений в таких подходах как оптимизация

муравьиными колониями или GRASP [Hoos and Stützle, 2005] также можно проинтерпретировать как разновидность поиска в окрестностях, но такая интерпретация не будет отображать базовые алгоритмические идеи этих методов [Birattari et al., 2001].

***По типу целевой функции.*** Некоторые алгоритмы модифицируют оценивание отдельных состояний процесса поиска во время выполнения алгоритма. Одним из примеров является метод “выламывания” (breakout) [Morris, 1993]. Базовой идеей является введение штрафов на включение отдельных компонент в решение, которые изменяют значение целевой функции. Как развитие этого подхода был предложен управляемый локальный поиск [Voudouris and Tsang, 1995]. Табу поиск также можно проинтерпретировать как такой, который использует динамическую целевую функцию, так как некоторые точки пространства поиска являются запрещенными, что отвечает бесконечно большим значениям целевой функции. Однако, остальные среди известных на данный момент методов используют статическую целевую функцию.

***По типу траектории.*** Важным отличием между разными итерационными подходами подходами является то, следуют ли они одной траектории поиска, что отвечает "непрерывным" переходам на графе соседства, или осуществляют большие "прыжки" на графе соседства [Birattari et al., 2001]. Предлагается такое формальное определение траекторно-непрерывного метода.

**Определение 4**. Метод называется траекторно-непрерывным, если последовательность текущих решений на итерациях $x_1, x_2, ..., x_k$ владеет таким свойством

$$\forall i = 2,...,k \; \exists N' \in \{N_1,...,N_l\} : x_i \in N'(x_{i-1}), \quad (1)$$

где $\{N_1,...,N_l\}$ – перечень систем окрестностей которые использует метод. Для популяционных траекторно-непрерывных методов должен существовать такой способ упорядочить решения из множеств текущих решений на итерациях в последовательности, что каждая из последовательностей будет владеть свойством (1). это свойство должно выполнятся для каждой последовательности текущих решений, отвечающих отдельным решениям из популяции. Методы не владеющие этим свойством называются траекторно-разрывными.

Имитационный отжиг или табу поиск являются типичными примерами траекторно-непрерывных методов. Алгоритмы, которые осуществляют более сложные переходы, но которые можно представить в виде более простых шагов, также можно проинтерпретировать как траекторно-непрерывных. К таким алгоритмам принадлежат, например, алгоритмы метода поиска переменной глубины [Hoos and Stützle, 2005], в частности, алгоритм Лина-Кернигана предложенный для задачи коммивояжера [Lin and Kernigan, 1973], или алгоритмы, которые базируются на "выбрасывании цепей" (ejection chains) [Glover, 1996]. В повторяемом локальном поиске, GRASP, меметических алгоритмах [Moscato, 1999] генерируются начальные точки для подчиненного локального поиска. Генерация начальных точек отвечает “прыжкам” в процессе поиска, и такие алгоритмы, в общем, выполняют траекторно-разрывными поиск по отношению к графу соседства, который используется в локальном поиске. Отметим, что практически все популяционные подходы являются траекторно-разрывными.

***По использованию памяти.*** Еще одной характеристикой, по которой будем различать приближенные алгоритмы КО является использование памяти. Алгоритмами с краткосрочной памятью называются алгоритмы, которые запоминают определенное количество последних принятых решений, сгенерированных решений, частей решений и т. д. Алгоритмами с долгосрочной памятью называются алгоритмы, которые запоминают информацию обо всем осуществленном процессе поиска в виде набора специальных переменных. Некоторые алгоритмы могут совмещать использование обоих видов памяти, например, табу-поиск с использованием критериев стремления [Glover and Laguna, 1997].

***По оценке точности.*** По оценке точности выделим алгоритмы с апостериорной оценкой точности, априорной оценкой и без соответствующих оценок. Апостериорные оценки вычисляют непосредственно во время процесса решения, учитывая при этом и само полученное решение. Априорная оценка точности – это гарантированная оценка, которая следует из самого алгоритма решения задачи.

***По сходимости.*** Для последнего показателя, по которому предлагается классифицировать приближенные методы комбинаторной оптимизации, – сходимости, в отличие от вышерассмотренных характеристик, принадлежность того или иного метода/алгоритма к классам построенным на основе этой характеристики может изменится в зависимости от получения новых результатов. Сходимость предусматривает два частных случая: сходимость по решению и сходимость по значению. Типы сходимости определяются таким образом [Dorigo and Blum, 2005].

**Определение 5**. Сходимостью по решению называется достижение с определенной вероятностью алгоритмом состояния, когда одно и то же оптимальное решение будет генерироваться постоянно.

**Определение 6**. Сходимостью по значению называется достижение с определенной вероятностью алгоритмом состояния, когда алгоритм сгенерирует произвольное оптимальное решение по крайней мере один раз.

Некоторые авторы [Birattari et al., 2001; Blum and Roli, 2003] предлагают разделять приближенные алгоритмы также по тому, были ли они созданы по какому-нибудь природному аналогу, как, например, имитационный отжиг или генетические алгоритмы [Hoos and Stützle, 2005]. Но этот показатель не отражает существенные характеристики алгоритма, поэтому в данную классификацию он включен не был.

## Заключение

Предложена классификация приближенных алгоритмов комбинаторной оптимизации. Для отдельных классов также предложена формализация их определения.

Сопоставление разработанной классификации с применяемыми на практике подходами показало, что, ввиду того, что все больше применяемых на практике алгоритмов являются гибридными, совмещая те или иные компоненты разных методов, нередко бывает невозможно четко классифицировать алгоритм в целом. Четкая классификация возможна только при разбиении алгоритм на отдельные процедуры, реализующие те или иные "типичные действия" (например, конструирование решений или возмущение решений). Это в свою очередь ставит вопрос определения перечня различных по функциональности элементарных процедур алгоритмов, который напрямую связан с построением общей (и в тоже время достаточно детальной) схемы приближенных методов комбинаторной оптимизации или, по крайней мере, достаточно широкого их класса. Это может быть одним из направлений будущих исследований.

## Список литературы

[Birattari et al., 2001] Birattari M., Paquete L., Stützle T., Varrentrapp K. (2001) Classification of metaheuristics and design of experiments for the analysis of components. Technical Report AIDA-01-05. Technische Universität Darmstadt. 12 p.

[Blum, 2003] Blum C., Roli A. (2003) Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. In: ACM Computing Surveys, 35, №3. P. 268–308.

[Dorigo and Blum, 2005] Dorigo M., Blum C. (2005) Ant colony optimization theory: A survey. In: Theoretical computer science. P. 243–278.

[Dorigo and Stützle, 2004] Dorigo M., Stützle T. (2004) Ant Colony Optimization. Cambridge: MIT Press, MA. 348 p.

[Drėo et al., 2007] Drėo J., Aumasson J.-P., Tfaili W., Siarry P. (2007) Adaptive Learning Search, A New Tool To Help Comprehending Metaheuristics. In: International Journal on Artificial Intelligence Tools. World Scientific Publishing Company, vol. 6, №3. P. 483-506.

[Glover, 1996] Glover F. (1996) Ejection Chains, Reference Structures and Alternating Path Methods for Traveling Salesman Problems. In:Discrete Applied Mathematics. №65. P. 223–253.

[Glover and Laguna, 1997] Glover F., Laguna M. (1997) Tabu Search. Kluwer Academic Publishers. 408 p.

[Hoos and Stützle, 2005] Hoos H.H., Stützle T. (2005) Stochastic Local Search: Foundations and Applications. San Francisco: Morgan Kaufmann Publ. 658 p.

[Kirkpatrik et al., 1983] Kirkpatrik S., Gelatt C. D., Vecchi M. P. (1983) Optimization by simulated annealing. In: Science. P. 671–680.

[Laguna, 2002] Laguna M. (2002) Global optimization and meta-heuristics. In: Encyclopedia of Life Support Systems, Theme 6.5, Topic 2. http://leeds-faculty.colorado.edu/Laguna/articles/elss.pdf.

[Lin and Kernighan, 1973] Lin S., Kernighan B.W. (1973) An effective heuristic algorithm for the travelling salesman problem In: Oper Res J. №21. P. 498–516.

[Lourenço et al., 2002] Lourenço H. R., Martin O., Stützle T. (2002) Iterated local search. In: Handbook of Metaheuristics: International Series in Operations Research & Management Science, vol. 57 (Eds. F. Glover and G. Kochenberger). Norwell: Kluwer Academic Publishers, MA. P. 321–353.

[Mladenović and Hansen, 1997] Mladenović N., Hansen P. (1997) Variable Neighbourhood Search. In: Computers & Operations Research. №24. P. 1097-1100.

[Morris, 1993] Morris P. (1993) The Breakout Method for Escaping from Local Minima. In: Proceeding of the 11th Conference on Artificial Intelligence MIT Press. P. 40-45.

[Moscato, 1999] Moscato P. (1999) Memetic Algorithms: A Short Introduction In: New Ideas in Optimization (Corne F. G. D., Dorigo M., Glover F.). McGraw-Hill. P. 219–234.

[Műhlenbein and Paaß, 1996] Műhlenbein H., Paaß G. (1996) From recombination of genes to the estimation of distributions I. Binary parameters. In: Lecture Notes in Computer Science: Parallel Problem Solving from Nature, №4. P. 178–187.

[Raidl, 2006] Raidl G. R. (2006) A Unified View on Hybrid Metaheuristics. In: Lecture Notes in Computer Science. Springer-Verlag. P. 1-12.

[Voudoris and Tsang,1995] Voudoris C., Tsang E. (1995) Guided Local Search. Technical Report CSM-247, Department of Computer Science, University of Essex, England.

[Гуляницкий, 2004] Гуляницкий Л.Ф. Решение задач комбинаторной оптимизации алгоритмами ускоренного вероятностного моделирования. В: Компьютерная математика. Сб.науч.тр, Киев, Институт кибернетики им. В.М. Глушкова НАН Украины, №1. С. 64-72.

[Гуляницкий, 2006] Гуляницкий Л.Ф. Об одном метаэвристическом методе комбинаторной оптимизации. В: Компьютерная математика. № 2. С. 1–6.

[Гуляницкий, 2008] Гуляницкий Л.Ф. (2008) До формалізації та класифікації задач комбінаторної оптимізації. В: Теорія оптимальних рішень. №7. С. 2–6.

[Пападимитриу и Стайглиц, 1985] Пападимитриу Х., Стайглиц К. (1985) Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир. 512 с.

## Об авторе

***Сергей Сиренко*** *(Sirenko) – аспирант, Институт кибернетики им. В.М.Глушкова НАН Украины, пр-т Глушкова, 40, Киев, 03680, Украина. E-mail: s.sirenko@gmail.com*