СИСТЕМНЫЙ АНАЛИЗ

И.В. СЕРГИЕНКО, Л.Ф. ГУЛЯНИЦКИЙ, С.И. СИРЕНКО

УДК 519.8

КЛАССИФИКАЦИЯ ПРИКЛАДНЫХ МЕТОДОВ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

Ключевые слова: *комбинаторная оптимизация, классификация методов, приближенные алгоритмы, метаэвристики, гиперэвристики.*

ВВЕДЕНИЕ

Развитие методов комбинаторной оптимизации (КО) после этапа первичного накопления знаний идет путем обобщения и систематизации полученных результатов. От рассмотрения отдельных методов КО и анализа их свойств переходят к исследованию концептуальных оснований и методологии построения агрегированных алгоритмов (метаэвристик, гиперэвристик), путей достижения адаптивности и повышенной эффективности при решении как конкретных задач КО, так и целых их классов.

В ряде работ рассматриваются вопросы классификации алгоритмов КО в целом и отдельных их классов [1–8]. В настоящей статье предлагается обобщение и развитие известных в литературе подходов к классификации. Ввиду большого количества предложенных методов и алгоритмов едва ли представляется возможным для всех существующих подходов к решению задач КО указать их место в классификации. Поэтому в работе упоминаются наиболее известные и широко используемые прикладные методы, которые более полно иллюстрируют соответствующий класс или характеристику, а также могут выступать в качестве технологических модулей при создании агрегированных алгоритмов. Отметим, что данная классификация, прежде всего, подчеркивает различия между алгоритмами, но при этом не лишает их общности.

ОПРЕДЕЛЕНИЕ ЗАДАЧИ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

Объектами, рассматриваемыми в задачах КО, обычно являются размещения, перестановки, подмножества, графы, целые числа и другие структуры, обобщением которых является понятие комбинаторного объекта [9]. Пусть заданы $Y=\{1,\dots,m\}$, Z — не более чем счетное пространство, которое назовем базовым, φ — гомоморфизм, $\varphi: Y \to Z$, удовлетворяющий ограничениям, заданным некоторым предикатом Ω.

Определение 1. Под комбинаторным объектом κ будем понимать триаду $\kappa=(\varphi, \mathrm{Z}, \Omega)$, а под его размерностью — мощность нумерующего множества Y.

Для определения задачи КО понадобится понятие локально-конечного пространства. Пусть $X=(X, d)$ — некоторое пространство с метрикой d. Обозначим O_ρ^d метрическую окрестность радиуса $\rho>0$: $O_\rho^d(x)=\{y \in X \mid d(x,y) \le \rho\}$.

Определение 2. Метрическое пространство X с метрикой d называется локально-конечным, если для любого $\rho \in (0, \infty)$ и $x \in X$ выполняется $|O_{\rho}^{d}(x)| < \infty$, где $|C|$ — мощность множества C.

Несложно убедиться, что любое конечное пространство X является локально-конечным при произвольном выборе метрики.

Определение 3. Задачей КО называется проблема нахождения такого $x_* \in D \subseteq X$, что

$$x_* = \arg \min_{x \in D \subseteq X} f(x), \tag{1}$$

где X — локально-конечное пространство решений задачи, элементами которого являются комбинаторные объекты, $D \subseteq X$ — подпространство допустимых решений, определяемое заданным предикатом Π, $f: X \to \mathbf{R}^1$ — целевая функция.

Отметим, что в общем случае пространство X может состоять из комбинаторных объектов разной размерности.

Определение 4. Система окрестностей, определенная на метрическом пространстве (X, d) решений задачи (1) — такое отображение $\mathbf{N}: X \to 2^{|X|}$, которое каждому решению $x \in X$ ставит в соответствие некоторое множество $N(x)$ такое, что $\forall x \in X$ $|N(x)| > 1$, $x \in N(x)$ и $\exists \rho \in (0, \infty): N(x) \subseteq O_{\rho}^{d}(x)$.

Множество $N(x)$ называется окрестностью варианта решения x, а его элементы называются соседними с x. Система окрестностей $\mathbf{N}$ определяет так называемый граф соседства $G_N = (X, V_N)$, $V_N = \{(x, y): y \in N(x)\}$.

Определение 5. Допустимое решение $x \in D$ называется локальным минимумом задачи (1) по отношению к системе окрестностей $\mathbf{N}$, если для произвольного $y \in N(x)$ выполняется неравенство $f(x) \leq f(y)$.

Определение 6. Если определена задача КО в смысле (1) и некоторая система окрестностей $\mathbf{N}$, то ландшафтом поиска называется тройка $(X, f, \mathbf{N})$.

На практике ряд алгоритмов работает не в пространстве X, а в некотором его расширении. Многие алгоритмы используют специальную оценочную функцию вместо целевой для управления процессом поиска. Эти два понятия формируют специальное представление задачи КО, которое является алгоритмо-ориентированным и вводится для адаптации постановки задачи к алгоритму ее решения. Поскольку такое представление задачи КО также можно рассматривать как отдельную задачу КО, если не будет оговорено противное, далее будем считать, что рассматриваемый алгоритм решает непосредственно задачу КО в виде (1).

ОБЩАЯ КЛАССИФИКАЦИЯ МЕТОДОВ КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

Рассмотрим алгоритм решения задач КО как процедуру, которая возвращает на выходе либо подмножество (возможно, пустое) допустимых решений, либо признак того, что задача неразрешима.

Алгоритмы КО можно классифицировать по многим характеристикам. Одной из самых важных является тип получаемого решения. По типу решения можно выделить точные, приближенные и эвристические алгоритмы.

Точные алгоритмы за конечное время гарантированно возвращают оптимальное решение или делают вывод, что его не существует, если задача неразрешима. Эти алгоритмы можно разделить на два класса: общие и специальные методы. Общие методы могут быть применены к весьма широкому кругу задач. Наибольшее распространение получили метод полного перебора, метод ветвей и границ, метод ветвей и сечений, последовательный анализ и отсеивание вариантов, динамическое программирование [10–13]. Специальные методы строятся для конкретных задач с учетом их специфики, как, например, метод Балаша («венгерский метод») для решения линейной задачи о назначениях.

Приближенные алгоритмы — это алгоритмы, гарантированно возвращающие вариант решения за конечное время (если оно существует), и для которых может быть получена оценка точности найденных вариантов решения. Оценки точности бывают двух типов: априорные и апостериорные. Априорная оценка точности — гарантированная оценка, задаваемая до начала решения задачи. Апостериорные оценки вычисляют непосредственно во время процесса решения или после его окончания, учитывая при этом и само полученное решение.

Под эвристическими алгоритмами («эвристиками») чаще всего понимают алгоритмы, для которых отсутствует либо неизвестна оценка точности. Это означает отсутствие гарантий того, что для конкретной решаемой задачи алгоритм не вернет как угодно «плохое» (в смысле целевой функции) решение или что он вернет решение вообще. В то же время эвристики корректны в том смысле, что не возвращают варианты решения, которые не являются допустимыми решениями задачи. Для многих практических задач такие алгоритмы показали свою эффективность, и более того, нередко эвристики могут быть единственным способом получить «хорошее» решение за приемлемое время.

Приближенными алгоритмами часто называют все «неточные» алгоритмы — как алгоритмы с оценкой точности, так и эвристические алгоритмы (используем это понятие в дальнейшем). Ввиду сложности многих практически важных задач КО применимость точных алгоритмов ограничена (например, задачами небольшой размерности или отдельных, сравнительно небольших классов) и маловероятно, что удастся разработать эффективные точные методы, применяемые к реальным задачам. Кроме того, схема точных алгоритмов часто не позволяет решать с их помощью некоторые типы задач КО, такие как динамические задачи или задачи с неопределенностями. Поэтому именно приближенные алгоритмы находят вся более широкое применение на практике, а их классификация будет главным предметом рассмотрения.

По сути, все вычислительные подходы к решению сложных задач КО могут быть описаны как поисковые процедуры, идея которых состоит в итерационном порождении/генерировании вариантов решений и их оценивании. Такой подход к описанию методов решения задач КО является общим, хотя и не отражает многих характеристик структуры, но в рамках данной работы его будет достаточно.

Приближенные алгоритмы часто строятся исходя из каких-либо правдоподобных идей (эвристик). На рис. 1 указаны основные парадигмы, используемые при создании прикладных приближенных методов КО, и приведены наиболее типичные примеры соответствующих методов.

Первые три класса имеют длительную историю применения и достаточно полно описаны в литературе [7, 10, 11, 14–17].

Эволюционные методы являются широким и разнообразным классом алгоритмов, использующих принципы биологической эволюции для решения оптимизационных задач [18, 19].

Роевой интеллект в контексте КО представлен классом алгоритмов, которые являются децентрализованными системами простых агентов, локально взаимодействующих со средой и между собой. Несмотря на отсутствие централизованной структуры управления агентами, локальное взаимодействие между ними приводит к проявлению глобального поведения всей системы в целом [20].

В методах сканирования пространства, принадлежащих к числу популяционных алгоритмов, на каждой итерации происходит формирование направления поиска в пространстве решений на основе нескольких имеющихся вариантов [7–9].

Довольно распространенным подходом к созданию приближенных алгоритмов КО является построение их вычислительной схемы на основе либо известных точных методов, либо максимального учета специфики конкретной задачи, что позволяет повысить эффективность, хотя и приводит к сужению применимости таких алгоритмов [10–12].

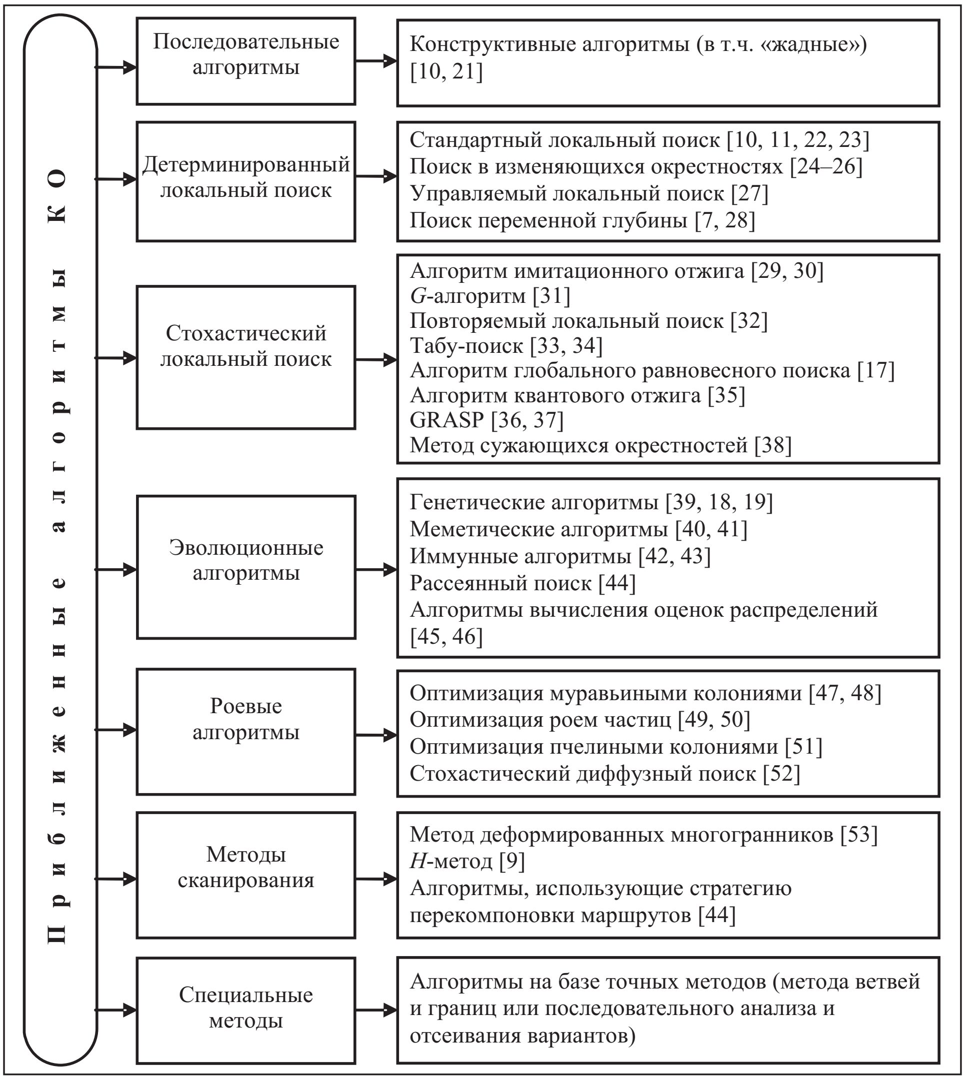

Рис. 1. Классификация алгоритмов КО по используемой парадигме

Предлагается классифицировать приближенные методы КО по таким характеристикам: принцип принятия решений; сложность структуры; тип используемых пространств; тип формируемой траектории; влияние на ландшафт поиска; использование памяти; наличие адаптации/обучения; наличие специальной модели задачи.

ПРИНЦИП ПРИНЯТИЯ РЕШЕНИЙ

В приближенных методах, прежде всего, будем различать по наличию рандомизации два принципа принятия решений — детерминированный и стохастический. Примером детерминированного метода является простой локальный поиск — алгоритм, который начиная с некоторого начального решения итерационно пытается его заменить лучшим (в смысле целевой функции) из окрестности [10, 11]. Один из наиболее известных стохастических алгоритмов — алгоритм имитационного отжига [29, 30]. Этот подход является развитием локального поиска — разрешается принятие в качестве текущего решения не только лучшего из окрестности, но и худшего по значению целевой функции решения. Выбор решения определяется вероятностным механизмом на основе значений целевой функции.

В настоящее время значительная часть разработанных и успешно применяемых на практике приближенных методов КО принадлежат именно к стохастическим [7, 14, 15, 17]. Такие методы показали свою эффективность при решении сложных прикладных задач КО.

СЛОЖНОСТЬ СТРУКТУРЫ

По сложности структуры будем различать простые алгоритмы, гибридные алгоритмы, метаэвристики, гибридные метаэвристики и гиперэвристки. Классификация по структуре весьма важна, но, поскольку, несмотря на широкое употребление понятий эвристики, метаэвристики и гиперэвристики, отсутствуют их общепринятые формальные определения, затруднительно четко разграничить алгоритмы согласно этой характеристике. Например, алгоритм имитационного отжига разные авторы относят как к метаэвристикам, так и к простым алгоритмам, учитывая то, что его отличие от локального поиска является незначительным, даже по сравнению с такой «простой» метаэвристической концепцией, как повторяемый локальный поиск [32]. Термин «метаэвристика» введен в работе [33], и упрощенно метаэвристические методы можно понимать как общую технику или подход, который используется для управления встроенными проблемно-зависимыми процедурами (методами) с целью повысить их эффективность или робастность [7]. Такие процедуры часто являются отдельными алгоритмами решения той же задачи, что и метод в целом. Метаэвристичекие методы описываются как общие подходы, которые конкретизацией проблемно-зависимых компонентов могут быть адаптированы к конкретной задаче или подклассу задач.

Под гибридными алгоритмами понимают такое объединение простых алгоритмов, которое еще не порождает метаэвристичный алгоритм. Например, это может быть простое последовательное или параллельное выполнение нескольких алгоритмов [54, 55]. На практике простые (как и гибридные) алгоритмы используются все реже, поскольку развитие метаэвристических подходов позволяет получать лучшие результаты, а темпы роста производительности вычислительных систем дают возможность применять все более сложные алгоритмы к задачам повышенной размерности. Исключение составляют случаи, когда удается разработать удачный специализированный алгоритм для отдельной задачи или узкого класса задач (эффективно учесть при разработке алгоритма особенности конкретной задачи), а также алгоритмы для задач повышенной размерности или задач с трудоемким вычислением целевой функции.

Гибридными метаэвристиками будем называть приближенные алгоритмы КО, которые комбинируют в себе компоненты из двух и более отдельных метаэвристик, точных методов, специализированных алгоритмов и т.д. К гибридным метаэвристикам также отнесем многоагентные системы, в которых агенты представляют отдельные алгоритмы (например, метаэвристики, точные или специализированные алгоритмы), взаимодействующие между собой. Классификация гибридных метаэвристик предложена и детально рассмотрена в работе [8]. По аналогии с последней возможно классифицировать гибридные алгоритмы по степени гибридизации, порядку выполнения и стратегии управления.

Гиперэвристика — метод, автоматизирующий процесс выбора, комбинирования, генерирования или адаптации ряда простых алгоритмов (эвристик) или их компонентов с целью эффективного решения задачи. Упрощенно под гиперэвристикой можно понимать (мета-)эвристики для выбора/конструирования (мета-)эвристик [56]. Одна из целей создания гиперэвристик — разработка оптимизационных систем, способных решать классы разных задач, а не только отдельные задачи. Детально гиперэвристики рассматриваются, в частности, в [56, 57].

ТИП ИСПОЛЬЗУЕМЫХ ПРОСТРАНСТВ

По такому критерию, прежде всего, выделим класс последовательных, или конструктивных алгоритмов. Их суть состоит в последовательном построении решения из отдельных частей по определенным правилам. Эти алгоритмы работают в некотором расширенном пространстве $S \supset X$. Во многих случаях такое пространство можно представить в терминах определений 1 и 3 в виде $S = X \cup \{\kappa^S = (\varphi^S, Z, \Omega) : (\varphi^S : Y_S \to Z \wedge Y_S \subset Y)\}$.

Простейшим примером последовательного алгоритма является эвристика, называемая «жадной», реализация которой для задачи коммивояжера известна как «иди в ближний (город)» [7, 21].

Другой важный класс алгоритмов — итерационные. Такие алгоритмы работают в пространстве решений исходной задачи. Итерационные методы разделяются по числу вариантов решений, которыми алгоритм одновременно оперирует [1] (генерирует и оценивает), на два класса: в случае одноэлементного множества алгоритмы называют одноточечными, а в случае, когда множество текущих вариантов решений содержит больше одного элемента, — популяционными. Одноточечные алгоритмы часто называют траекторными, поскольку процесс их работы можно представить в виде непрерывной траектории (пути) на графе соседства (понятие траекторной непрерывности рассматривается ниже). Алгоритмы стохастического локального поиска, такие как, например, имитационный отжиг, управляемый локальный поиск [27] и GRASP [37], являются одноточечными. Название «популяционные алгоритмы» введено по аналогии с терминологией, используемой в эволюционных методах. Эволюционные методы, такие как генетические алгоритмы [39], были одними из первых, оперировавших множеством вариантов решений одновременно. К числу ранних популяционных алгоритмов решения задач КО относятся и фронтальные алгоритмы [58].

Конструктивные методы обычно являются наиболее быстрыми методами решения задач КО, но, в общем случае, проигрывают итерационным методам в «качестве» решений. На практике конструктивные алгоритмы чаще всего используются либо для задач повышенной размерности или задач с затратным вычислением целевой функции, либо для генерирования начальных решений для других методов. Иногда они встроены в метод более сложным образом. Например, в оптимизации муравьиными колониями [48] конструктивный блок — деятельность искусственных муравьев — играет ключевую роль.

ТИП ФОРМИРУЕМОЙ ТРАЕКТОРИИ

Важным отличием между разными итерационными методами является то, формируют ли они одну траекторию поиска, что отвечает «непрерывным» переходам на графе соседства, или осуществляют большие «прыжки» на этом графе [3]. Предлагается следующее формальное определение траекторно-непрерывного метода.

Определение 7. Пусть дана совокупность систем окрестностей $\{N_1,\dots,N_L\}$ и алгоритм, оперирующий на каждой итерации i множеством вариантов решений X_i. Алгоритм называется траекторно-непрерывным относительно систем окрестностей $\{N_1,\dots,N_L\}$, если для всех $i \geq 2$ выполняется

$$X_i \subseteq \bigcup_{j=1}^{L} \bigcup_{x_{i-1} \in X_{i-1}} N_j(x_{i-1}). \tag{2}$$

Если множество X_i состоит из одного элемента x_i (например, алгоритм является одноточечным), то условие (2) эквивалентно условию $\exists N^i \in \{N_1,\dots$ $\dots,N_L\} : x_i \in N^i(x_{i-1})$.

Определение 8. Алгоритмы, не удовлетворяющие условию (2), называются траекторно-разрывными относительно систем окрестностей $\{N_1,\dots,N_L\}$.

Траекторно-непрерывность алгоритма относительно систем окрестностей $\{N_1,\dots,N_L\}$ означает, что для каждого сгенерированного алгоритмом решения $x \in X_i$ существует путь в агрегированном графе соседства $G_{N_1,\dots,N_L} =$ $= (X, \bigcup_{i \in \{1,\dots,L\}} V_{N_i})$, который соединяет некоторое начальное решение и x. В случае одноточечного траекторно-непрерывного метода процесс поиска является построением траектории (пути) на соответствующем агрегированном графе соседства.

Имитационный отжиг и табу-поиск [34] — типичные примеры траекторно-непрерывных методов. Алгоритмы, осуществляющие более сложные переходы, но которые представимы в виде более простых шагов, также можно интерпретировать как траекторно-непрерывные [1]. К таким алгоритмам относятся, например, алгоритмы метода поиска переменной глубины [7], в частности алгоритм Лина–Кернигана [28], или алгоритмы, которые базируются на «выбрасывании цепей» (ejection chains) [59].

Большинство популяционных алгоритмов являются траекторно-разрывными относительно любой одной фиксированной системы окрестностей. Среди исключений — фронтальный алгоритм. В модификации этого алгоритма, которая базируется на локальном поиске, на каждой итерации осуществляются параллельные локальные улучшающие шаги относительно некоторой наперед заданной системы окрестностей [58].

Понятие траекторно-непрерывности алгоритма также связано с количеством систем окрестностей, которые используются в методе: алгоритмы, использующие больше одной системы окрестностей, будут траекторно-разрывными относительно любой одной системы окрестностей.

ВЛИЯНИЕ НА ЛАНДШАФТ ПОИСКА

Выделим в отдельный класс методы КО, которые изменяют ландшафт поиска в процессе своей работы. Возможны следующие типы изменений: пространства поиска, целевой/оценочной функции и системы окрестностей. Рассмотрим каждый из них в отдельности.

Изменение пространства поиска. Под изменением пространства поиска понимается включение/исключение элементов в/из него. В табу-поиске запрещено возвращение к недавно рассмотренным вариантам решения, что соответствует их временному исключению из пространства поиска. Большинство методов КО не изменяют пространство поиска.

Изменение целевой/оценочной функции. Некоторые алгоритмы модифицируют оценивание отдельных состояний процесса поиска во время выполнения алгоритма [1]. Одним из примеров является метод «выламывания» (breakout) [60], базовой идеей которого является введение штрафов на включение отдельных компонентов в решение, изменяющих значение целевой функции. В качестве развития этого подхода был предложен управляемый локальный поиск [27].

Табу-поиск можно интерпретировать как такой, который использует динамическую целевую функцию: в алгоритме некоторые точки пространства поиска запрещаются, что равноценно присвоению бесконечно больших значений оценочной функции. Но такая интерпретация отличается от базовой алгоритмической идеи метода.

Изменение системы окрестностей. Многие из приближенных алгоритмов КО используют в процессе поиска одну систему окрестностей. Это, например, такие алгоритмы, как имитационный отжиг, табу-поиск, G-алгоритмы [31]. В повторяемом локальном поиске используется по крайней мере две системы окрестностей: вначале осуществляется локальный поиск до достижения локального минимума, после этого — возмущение найденного локального минимума. Фактически, возмущение может рассматриваться как шаг в другой системе окрестностей, отличной от той, по отношению к которой был найден локальный минимум. Эти соображения, в частности, развиты независимо в методах поиска в пульсирующих окрестностях [24] и поиска в изменяемых окрестностях [25, 26] — в этих подходах используемые окрестности систематически изменяются в пределах заведомо заданного их перечня.

ИСПОЛЬЗОВАНИЕ ПАМЯТИ

Очень важной характеристикой, по которой будем различать приближенные алгоритмы КО, является использование памяти. Память — это набор специальных переменных, в которых отображается опыт (информация), накопленный в процессе поиска (работы алгоритма). В методах КО имеется несколько функций памяти. Во-первых, память может использоваться для хранения информации, которая будет использована на выходе алгоритма (например, наилучшее найденное решение). Во-вторых, память может использоваться для управления процессом поиска. Этот тип памяти может быть кратковременным или долговременным. Кратковременная память подразумевает запоминание определенного количества последних сгенерированных решений, частей решений и принятых решений. Запоминание в табу-поиске последних сгенерированных решений является примером кратковременной памяти. Долговременная память — набор специальных дополнительных переменных, в которых запоминается информация обо всем осуществленном процессе поиска. Величины штрафов в алгоритме управляемого локального поиска являются примером долговременной памяти. Функция управления поиском тесно связана с влиянием на ландшафт поиска: алгоритмы с этой функцией памяти изменяют ландшафт поиска. Последняя и наиболее важная функция памяти — адаптация.

НАЛИЧИЕ АДАПТАЦИИ/ОБУЧЕНИЯ

Среди приближенных методов КО с памятью выделим такие, которые обладают свойством адаптироваться/обучаться (изменять значения своих параметров в процессе работы на основе состояния памяти). Адаптация включает динамическую настройку значений параметров, автоматический выбор подчиненных процедур и наличие модели задачи. Адаптивные методы КО — это новая развивающаяся область, и пока она является скорее набором отдельных подходов. Рассмотрим некоторые из них.

Реактивный/реагирующий поиск [61] предоставляет механизм для включения в алгоритм настройки варьируемых параметров: значения параметров настраиваются в автоматизированном цикле обратной связи соответственно «качеству» найденных решений, истории поиска и другим характеристикам. В реактивном табу-поиске [62], одном из первых реактивных алгоритмов, период запрещения автоматически настраивается в процессе поиска.

Процесс обучения активно используется для автоматизированного выбора компонентов, осуществляемого в алгоритмах, которые принадлежат к описанному классу гиперэвристик.

Адаптация может достигаться путем использования специальной структуры памяти — модели задачи, о чем пойдет речь далее.

Конкретные техники адаптации развиваются и в рамках отдельных классов методов. Например, в классе меметических алгоритмов [41] развивается направление адаптивных методов [63], в которых совершается динамический выбор типа алгоритма встроенного локального поиска.

Рассмотренные подходы к разработке адаптивных алгоритмов КО, естественно, не охватывают всех предложенных методов и техник, а лишь отображают наиболее значимые тенденции.

НАЛИЧИЕ СПЕЦИАЛЬНОЙ МОДЕЛИ ЗАДАЧИ

По отсутствию или наличию в алгоритме специальной модели задачи можно выделить класс задаче-ориентированных методов и класс моделе-ориентированных алгоритмов [64]. Большинство методов КО являются задаче-ориентированными, так как они генерируют новые варианты решений только на основе текущего решения или множества текущих вариантов решений. Это такие методы, как генетические алгоритмы, повторяемый локальный поиск и др. В моделе-ориентированных методах варианты решений генерируются с использованием параметризированной вероятностной модели, которая обновляется на основе ранее рассмотренных вариантов решений так, чтобы поиск концентрировался в областях с вариантами решений «высокого качества» [64]. Широко известными моделе-ориентированными методами являются метод оптимизации муравьиными колониями [48], метод кросс-энтропии [65] и метод вычисления оценок распределений [46].

В [66] предложено в одном методе использовать несколько встроенных многоагентных моделе-ориентированных алгоритмов, по результатам работы которых на каждой итерации общего метода формируется и возвращается встроенным методам для использования обобщенная/агрегированная модель задачи.

ДРУГИЕ ПОДХОДЫ К КЛАССИФИКАЦИИ

Некоторые авторы [1, 4] предлагают разделять приближенные алгоритмы по тому, были ли они созданы по какому-нибудь природному аналогу, как, например, алгоритмы оптимизации муравьиными колониями или генетические алгоритмы. Однако этот показатель не отражает существенных характеристик алгоритма, поэтому в данную классификацию включен не был.

На рис. 2 приведены важнейшие характеристики, по которым предлагается классифицировать прикладные приближенные алгоритмы КО, а также соответствующие классы, образованные на основе этих характеристик. Кроме того, на рисунке отображены взаимосвязи между классами.

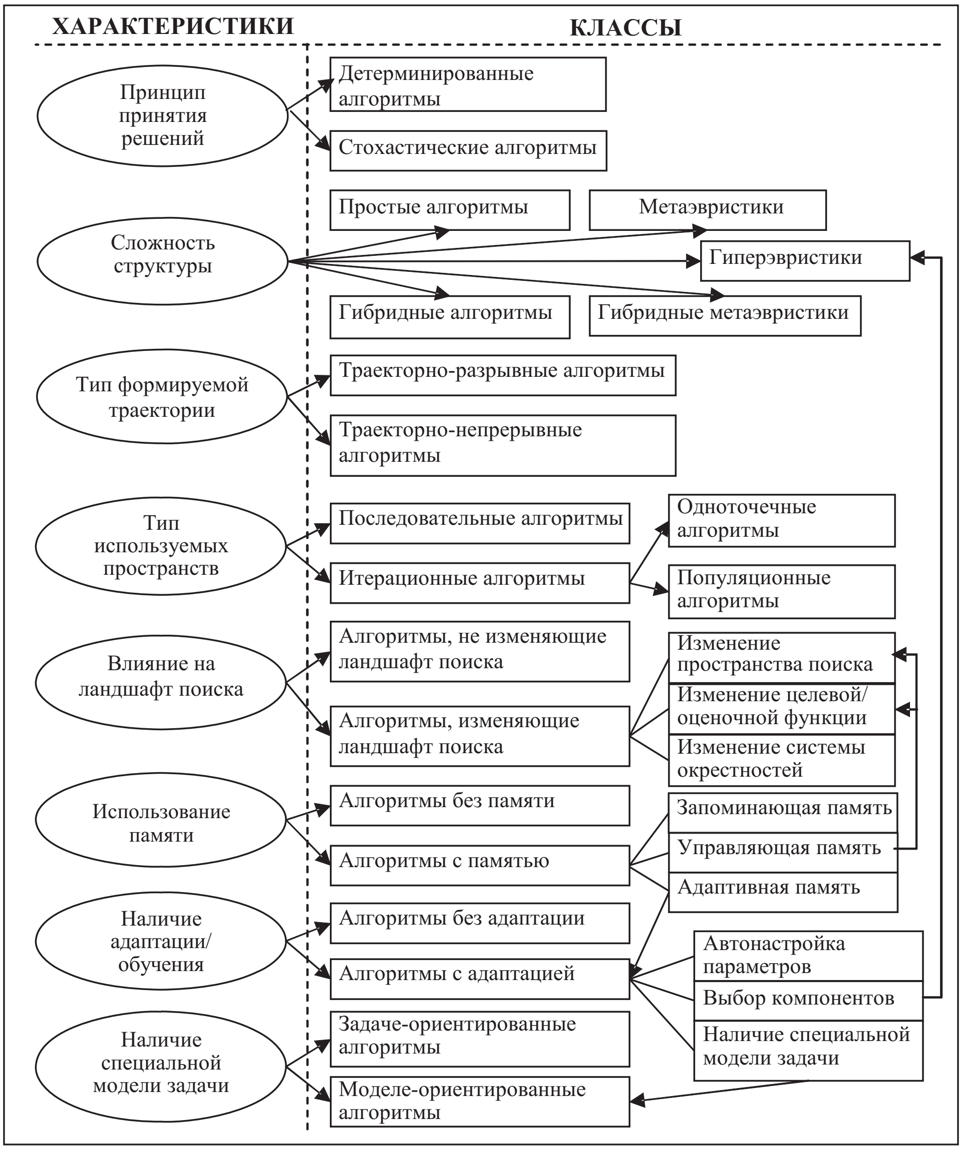

Рис. 2. Классификация прикладных приближенных алгоритмов КО

Аспектам параллельной реализации прикладных алгоритмов комбинаторной оптимизации, рассмотрение которых находится вне рамок данной работы, уделено внимание, в частности, в [67, 68].

Отметим, что в данной работе рассматриваются задачи КО в классической постановке, т.е. вне поля зрения были оставлены другие типы задач КО (например, дискретное программирование, многокритериальные, динамические, стохастические задачи).

В последнее время появился ряд новых парадигм, на основе которых разрабатываются алгоритмы КО [69–74]. Однако имеющихся данных об их применении для решения практически важных задач КО еще недостаточно для того, чтобы сделать обоснованные выводы об их эффективности.

ЗАКЛЮЧЕНИЕ

Предложены подходы к формализации задач КО и классификации прикладных методов КО. По используемой парадигме среди приближенных алгоритмов решения задач КО выделены конструктивные алгоритмы, алгоритмы детерминированного локального поиска, стохастического локального поиска, эволюционные методы, методы роевого интеллекта, методы сканирования и специальные алгоритмы. Сформулированы критерии, по которым возможно осуществлять разностороннюю классификацию большинства известных и практически применяемых приближенных алгоритмов решения задач КО: принцип принятия решений, сложность структуры, тип используемых пространств, тип формируемой траектории, влияние на ландшафт поиска, использование памяти, наличие адаптации/обучения и наличие специальной модели задачи.

Реализуемые в настоящее время алгоритмы решения практических задач КО в подавляющем большинстве являются (гибридными) метаэвристиками, т.е. объединяют в себе идеи и компоненты из нескольких подходов. Данная классификация рассматривает концептуальные идеи построения «базовых» прикладных алгоритмов КО, что позволяет эффективно применять и анализировать технологии создания современных гибридных метаэвристик и гиперэвристик.

СПИСОК ЛИТЕРАТУРЫ

1. Stützle T. Local search algorithms for combinatorial problems — analysis, improvements and new applications: PhD. Thesis / Technische Univ. Darmstadt. — Darmstadt, 1998. — 214 p.
2. Vaessens R.J.M., Aarts E.H.L., Lenstra J.K. A local search template // Comput. Oper. Res. — 1998. — **25**, N 11. — P. 969–979.
3. Birattari M., Paquete L., Stützle T., Varrentrapp K. Classification of metaheuristics and design of experiments for the analysis of components: (Techn. rep.) / Techn. Univ. Darmstadt. — N AIDA-01-05. — Darmstadt, 2001. — 12 p.
4. Blum C., Roli A. Metaheuristics in combinatorial optimization: overview and conceptual comparison // ACM Computing Surveys. — 2003. — **35**, N 3. — P. 268–308.
5. Gendreau M., Potvin J.-Y. Metaheuristics in combinatorial optimization // Ann. Oper. Res. — 2005. — **140**, N 1. — P. 189–213.
6. Леонтьев В.К. Дискретная оптимизация // Журн. вычисл. математики и мат. физики. — 2007. — **47**, № 2. — С. 338–352.
7. Hoos H.H., Stützle T. Stochastic local search: foundations and applications. — San Francisco: Morgan Kaufmann Publ., 2005. — 658 p.
8. Raidl G.R. A unified view on hybrid metaheuristics // Lect. Notes Comput. Sci. — Berlin: Springer-Verlag, 2006. — **4030**. — P. 1–12.
9. Гуляницкий Л.Ф., Сергиенко И.В. Метаэвристический метод деформированного многогранника в комбинаторной оптимизации // Кибернетика и системный анализ. — 2007. — № 6. — С. 70–79.

10. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. — М.: Мир, 1985. — 512 с.
11. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации. — К.: Наук. думка, 1988. — 471 с.
12. Михалевич В.С., Кукса А.И. Методы последовательной оптимизации в дискретных сетевых задачах оптимального распределения ресурсов. — М.: Наука, 1983. — 208 с.
13. Павлов А.А., Мисюра Е.Б. Эффективный точный ПДС-алгоритм решения задачи о суммарном запаздывании для одного прибора // Системні дослідження та інформ. технології. — 2004. — № 4. — С. 30–59.
14. Handbook of applied optimization (Eds. P. Pardalos, M.G.C. Resende). — Oxford: Oxford Univ. Press, 2002. — 2026 p.
15. Handbook of approximation algorithms and metaheuristics (Ed. T.F. Gonzalez). — Boca Raton (FL): CRC press, 2007. — 1432 p.
16. Сергиенко И.В., Каспшицкая М.Ф. Модели и методы решения на ЭВМ комбинаторных задач оптимизации. — К.: Наук. думка, 1981. — 288 с.
17. Сергиенко И.В., Шило В.П. Задачи дискретной оптимизации. — К.: Наук. думка, 2003. — 261 с.
18. De Jong K.A. Evolutionary computation: a unified approach. — Cambridge (MA): MIT Press, 2006. — 272 p.
19. Leguizamon G., Blum C., Alba E. Evolutionary computation // Handbook of approximation algorithms and metaheuristics (Ed. T.F. Gonzalez). — Boca Raton (FL): CRC press, 2007. — P. 372–386.
20. Kennedy J., Eberhart R.C., Shi Y. Swarm intelligence. — San Francisco (CA): Morgan Kaufmann, 2001. — 512 p.
21. Khuller S., Raghavachari B., Young N.E. Greedy methods // Handbook of approximation algorithms and metaheuristics (Ed. T.F. Gonzalez). — Boca Raton (FL): CRC press, 2007. — P. 67–80.
22. Сергієнко І.В. Один метод розв’язування задач на відшукання екстремальних значень // Автоматика. — 1964. — № 5. — С. 15–21.
23. Кочетов Ю.А. Вычислительные возможности локального поиска в комбинаторной оптимизации // Журн. вычисл. математики и мат. физики. — 2008. — **48**, № 5. — С. 788–807.
24. Гуляницкий Л.Ф., Ходзинский А.Н. Особенности реализации алгоритмов метода ветвей и границ и метода вектора спада в пакете ВЕКТОР–1В // Вычислительные аспекты в пакетах прикладных программ. — Киев: Ин-т кибернетики АН УССР, 1979. — С. 25–30.
25. Mladenović N., Hansen P. Variable neighbourhood search // Comput. Oper. Res. — 1997. — N 24. — P. 1097–1100.
26. Hansen P., Mladenović N., Moreno-Pérez J.A. Variable neighbourhood search: methods and applications // Quarterly J. Oper. Res. — 2008. — **6**, N 4. — P. 319–360.
27. Voudouris C., Tsang E.P.K. Guided local search // Handbook of metaheuristics (Ed. F. Glover). — Norwell (MA): Kluwer Acad. Publ., 2003. — P. 185–218.
28. Kernighan B.W., Lin S. An efficient heuristic procedure for partitioning graphs // Bell System Techn. J. — 1970. — N 49. — P. 291–307.
29. Kirkpatrik S., Gelatt C.D., Vecchi M.P. Optimization by simulated annealing // Science. — 1983. — **220**, N 4598. — P. 671–680.
30. Aarts E., Korst J., Michiels W. Simulated annealing // Handbook of approximation algorithms and metaheuristics (Ed. T.F. Gonzalez). — Boca Raton (FL): CRC press, 2007. — P. 387–397.
31. Гуляницкий Л.Ф. Решение задач комбинаторной оптимизации алгоритмами ускоренного вероятностного моделирования // Компьютерная математика. — К.: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 2004. — № 1. — С. 64–72.
32. Lourenço H.R., Martin O., Stützle T. Iterated local search // Handbook of metaheuristics (Eds. F. Glover, G. Kochenberger). — Norwell (MA): Kluwer Acad. Publ., 2002. — P. 321–353.

33. Glover F. Future paths for integer programming and links to artificial intelligence // Comput. Oper. Res. — 1986. — N 5. — P. 533–549.
34. Glover F., Laguna M. Tabu search. — Norwell (MA): Kluwer Acad. Publ., 1997. — 408 p.
35. Quantum annealing and related optimization methods (Eds. A. Das, B.K. Chakrabarti) // Lect. Notes Physics. — Heidelberg: Springer, 2005. — **679**. — 377 p.
36. Feo T.A., Resende M.G.C. A probabilistic heuristic for a computationally difficult set covering problem // Oper. Res. Letters. — 1989. — **8**, N 2. — P. 67–71.
37. Pitsoulis L., Resende M.G.C. Greedy randomized adaptive search procedures // Handbook of applied optimization (Eds. P.M. Pardalos, M.G.C. Resende). — Oxford: Oxford Univ. Press, 2002. — P. 168–181.
38. Стоян Ю.Г., Яковлев С.В. Математические модели и оптимизационные методы геометрического проектирования. — К.: Наук. думка, 1986. — 268 с.
39. Holland J.H. Adaptation in natural and artificial systems. — Ann Arbor (MI): Univ. of Michigan Press, 1975. — 183 p.
40. Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms: (Techn. Rep.) / California Inst. of Technology. — N C3P 826. — Pasadena (CA), 1989. — 68 p.
41. Moscato P., Cotta C. Memetic algorithms // Handbook of approximation algorithms and metaheuristics (Ed. T.F. Gonzalez). — Boca Raton (FL): CRC press, 2007. — P. 412–423.
42. Cutello V., Nicosia G. An immunological approach to combinatorial optimization problems // Lect. Notes Comput. Sci. — 2002. — N 2527. — P. 361–370.
43. de Castro L.N., Timmis J. Artificial immune systems: a new computational intelligence approach. — London: Springer, 2002. — 380 p.
44. Glover F., Laguna M., Martí R. Scatter search and path relinking: foundations and advanced designs // New optimization techniques in engineering (Eds. G. Onwubolu, B.V. Babu). — Berlin: Springer-Verlag, 2004. — P. 87–100.
45. Mühlenbein H., Paaβ Г. From recombination of genes to the estimation of distributions. I. Binary parameters // Lect. Notes Comput. Sci.: Parallel problem solving from nature. — 1996. — N 4. — P. 178–187.
46. Estimation of distribution algorithms. A new tool for evolutionary computation (Eds. P. Larrañaga, J.A. Lozano). — Boston (MA): Kluwer Acad. Publ., 2001. — 416 p.
47. Dorigo M., Di Caro G., Gambardella L.M. Ant algorithms for discrete optimization // Artif. Life. — 1999. — **5**, N 2. — P. 137–172.
48. Dorigo M., Stützle T. Ant colony optimization. — Cambridge (MA): MIT Press, 2004. — 348 p.
49. Eberhart R.C., Kennedy J. Particle swarm optimization // Proc. IEEE Intern. Conf. on Neural Networks. — Piscataway (NJ): IEEE Service Center, 1995. — **4**. — P. 1942–1948.
50. Clerc M. Particle swarm optimization. — Hoboken (NJ): Wiley-Interscience, 2006. — 243 p.
51. Teodorovic D., Lucic P., Markovic G., D'Orco M. Bee colony optimization: principles and applications // 8th Seminar on Neural Network Applications in Electrical Engineering, Belgrade, Serbia, 2006. — P. 151–156.
52. De Meyer K., Nasuto S.J., Bishop J.M. Stochastic diffusion search: partial function evaluation in swarm intelligence dynamic optimisation // Stigmergic optimization (Eds. A. Abraham, C. Grosam, V. Ramos). — Berlin: Springer–Verlag, 2006. — P. 185–208.
53. Гуляницкий Л.Ф. Метод деформаций в дискретной оптимизации // Исслед. операций и АСУ. — 1989. — № 34. — С. 30–33.
54. Журавлев Ю.И. Корректные алгебры над множествами некорректных (эвристических) алгоритмов // Кибернетика. — 1977. — № 4. — С. 5–17.
55. Bertsekas D.P., Tsitsiklis J.N., Wu C. Rollout algorithms for combinatorial optimization // J. Heuristics. — 1997. — N 3. — P. 245–262.
56. Hyperheuristics: an emerging direction in modern search technology / E. Burke, E. Hart, G. Kendall et al. // Handbook of metaheuristics (Eds. F. Glover, G. Kochenberger). — Norwell (MA): Kluwer Acad. Publ., 2003. — P. 457–474.

57. Ozcan E., Bilgin E., Korkmaz E.E. A comprehensive analysis of hyper-heuristics // Intelligent Data Analysis. — 2008. — **12**, N 1. — P. 3–23.
58. Гуляницкий Л.Ф. Об одном семействе итерационных алгоритмов дискретной оптимизации // Разработка математических и технических средств в АСУ. — Киев: ИК АН УССР, 1978. — С. 25–30.
59. Glover F. Ejection chains, reference structures and alternating path methods for traveling salesman problems // Discrete Appl. Math. — 1996. — N 65. — P. 223–253.
60. Morris P. The breakout method for escaping from local minima // Proc. of the 11th Conf. on Artif. Intelligence. — Cambridge (MA): MIT Press, 1993. — P. 40–45.
61. Battiti R., Brunato M. Reactive search: Machine learning for memory-based heuristics: (Techn. Rep.) / Univ. di Trento. — N DIT-05-058. — Trento, 2005. — 20 p.
62. Battiti R., Tecchiolli R. The reactive tabu search // ORSA J. Computing. — 1994. — **6**, N 2. — P. 126–140.
63. Ong Y.-S., Lim M.-H., Zhu N., Wong K.-W. Classification of adaptive memetic algorithms: a comparative study // IEEE Trans. Systems, Man, and Cybernetics. Part B, Cybernetics. — 2006. — **36**, N 1. — P. 141–152.
64. Zlochin M., Birattari M., Meuleau N., Dorigo M. Model-based search for combinatorial optimization: a critical survey // Ann. Oper. Res. — 2004. — N 131. — P. 373–395.
65. Rubinstein R.Y., Kroese D.P. The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning. — New York: Springer-Verlag, 2004. — 300 p.
66. Гуляницький Л.Ф. Розробка кооперативних метаевристик // Abstract of Int. Conf. «Problems of Decision Making under Uncertainties (PDMU–2009)» (April 27-30, 2009, Skhidnytsia, Ukraine). — Kyiv, 2009. — P. 90–91.
67. Parallel combinatorial optimization (Ed. E.-G. Talbi). — Hoboken (NJ): Wiley-Interscience, 2006. — 330 p.
68. Parallel metaheuristics: a new class of algorithms (Ed. E. Alba). — Hoboken (NJ): Wiley-Interscience, 2006. — 576 p.
69. Choi C., Lee J. Chaotic local search algorithm // Artif. Life Robotics. — 1998. — **2**, N 1. — P. 41–47.
70. Boettcher S., Percus A.G. Extremal optimization: Methods derived from co-evolution // Proc. of the Genetic and Evolutionary Computation Conf. (GECCO), 1999. — P. 825–832.
71. Bourjot C., Chevrier V., Thomas V. A new swarm mechanism based on social spiders colonies: from web weaving to region detection // Web Intelligence and Agent Systems. — 2003. — **1**, N 1. — P. 47–64.
72. Wakuya H. A new search method for combinatorial optimization problem inspired by the spin glass system // Intern. Congress Series. — 2006. — N 1291. — P. 201–204.
73. Cortés P., García J.M., Muñuzuri J., Onieva L. Viral systems: A new bio-inspired optimisation approach // Comput. Oper. Res. — 2008. — N 35. — P. 2840–2860.
74. Omran M.G.H., Mahdavi M. Global-best harmony search // Appl. Math. Comput. — 2008. — N 198. — P. 643–656.

Поступила 29.04.2009