

В. Я. Цветков, д-р техн. наук, проф., зам. руководителя Центра перспективных фундаментальных и прикладных исследований,
Научно-исследовательский и проектно-конструкторский институт информатизации, автоматизации и связи на железнодорожном транспорте (НИИАС),
Москва, 109029, Россия, e-mail: cvj2@mail.ru

Алгоритмы как средство познания

Исследуется содержание алгоритмов первого и второго рода. Рассмотрено шесть типовых алгоритмов первого рода. Показано сходство и различие между алгоритмами. Определен критерий отнесения алгоритма к первому или второму роду. Показано, что главным разграничением алгоритмов являются когнитивные признаки обзорности и воспринимаемости. Дан анализ алгоритмов второго рода и определены методы их построения.

Ключевые слова: вычисление, познание, алгоритм, линейный алгоритм, прямой алгоритм, алгоритм первого рода, алгоритм второго рода, воспринимаемость задачи

Введение

Современное понятие алгоритма можно рассматривать не только как описание пути вычисления, но и как средства познания и передачи знаний. Опыт решения задач с применением алгоритмов показал, что по топологии решения можно выделить два типа задач и соответствующие им два типа алгоритмов. В одном типе алгоритмов, его называют алгоритм первого рода, решение удается получить от постановки задачи до окончательного решения за один проход или за один цикл. Этот тип алгоритма широко применяется в вычислительных системах.

В другом типе алгоритмов, его называют алгоритм второго рода, решение за один проход или за один цикл получить в принципе невозможно. Оно получается последовательно путем частичного решения задачи и последующим выбором дальнейшего частичного решения и затем выбором нового этапа и т. д. до получения окончательного решения, которое удовлетворяет начальным условиям.

Ярким примером задач и алгоритмов второго рода является шахматная партия. В начальном состоянии шахматной игры невозможно сказать, кто из соперников победит и чем закончится данная партия. Это невозмож-

но сказать наблюдателю со стороны и каждому из игроков. Когда игроки начинают делать первые ходы, ситуация проясняется для каждого из игроков и для стороннего наблюдателя. Каждый ход шахматной партии можно рассматривать как цикл решения задачи. Чем больше ходов, тем ближе окончание партии, т. е. решение данной проблемной ситуации. Значение результата шахматной игры является частным для каждого из игроков и общим для опыта игры и накопления знаний о методах игры. Точно также решение задач второго рода имеет частное и общее значение.

По существу, решение задач второго рода описывает решение сложных проблем и позволяет накапливать опыт в решении проблем и познания мира. Именно в таком аспекте рассматривается алгоритм в монографии академика Н. Моисеева "Алгоритмы развития" [1]. Н. Моисеев использует термин "алгоритм" в расширительном смысле, не как вычислительный механизм, а как средство решения сложных проблем и познания мира.

Другой относительно новый термин, используемый в настоящей работе, — это "экстернализация". Он означает преобразование неявного неформализованного знания в явное формализованное знание. Этот термин введен японским ученым Нонака, в его известной работе [2] (бо-

более 22 000 цитирований), посвященной преобразованию неявного знания в явное знание. Идеи трансформации знания развиты в работе [3].

С позиций экстернализации знания алгоритм — это средство для решения такой задачи. Алгоритм представляет собой преобразование одной формы знания (условие задачи) в другую форму знаний (решение задачи). В своей работе Нонака описывает четырехступенчатый алгоритм, который решает четыре качественно разные задачи.

В обычных вычислениях алгоритм решает одну качественную задачу: он преобразует начальные условия в решение. При этом, как правило, это либо алгоритм только первого рода — прямое решение, либо алгоритм второго рода — поэтапное решение.

В четырехступенчатой схеме [2] Нонака описывает совместно алгоритмы первого и второго рода, хотя такие термины не употребляет. Альтернативой точке зрения автора работы [1] является рассмотрение алгоритма как технологической компоненты при обработке информации. Эта точка зрения отражена в известной работе Т. Кормена [4], выпущенной в 1990 г., многократно переизданной [5]. В данной работе заложены канонические основы построения только вычислительных алгоритмов первого рода. Первым базовым принципом построения алгоритмов по Т. Кормену представлен эвристический принцип "разделяй и властвуй". По сути дела, это алгоритм второго рода, использующий качественный анализ. То есть для построения алгоритма первого рода Т. Комен использует алгоритм второго рода.

По мнению автора данной статьи многие современные алгоритмы можно рассматривать также как сложные системы, поскольку они обладают признаками сложных систем. Алгоритм можно отнести к классу закрытых детерминированных сложных систем и рассматривать как функциональную систему, поскольку он всегда выполняет определенную функцию. Функциональное описание сложных систем исследовано М. Месаровичем [6], поэтому можно использовать предложенный им формализм для описания алгоритмов.

Алгоритмы имеют устойчивые связи и относятся к системам с постоянной структурой. Описание алгоритма можно рассматривать как явное знание, которое воспринимается разными людьми и может передаваться от одного человека к другому. Совокупность алгоритмов, решающих родственные задачи, дает возможность делать научное обобщение, которое вы-

ходит за рамки одного технического решения задачи, также как отдельная шахматная партия дает основание обобщать теорию игры в шахматы. Это дает основание связывать алгоритмы с получением знания.

Общеизвестно, что люди потенциально способны воспринимать примерно одинаковые объемы информации. Однако выводы, которые делают субъекты на основе одинаковых порций информации, получаются разными. Это создает противоречия и необходимость поиска компромиссов. В отличие от субъектов, алгоритмы на основе одинаковых порций информации получают одинаковые результаты. С этих позиций алгоритмы можно рассматривать как механизм, который способен найти компромиссы и разрешить противоречия.

Формализация алгоритмов первого рода

Построение алгоритмов связано с функциональными преобразованиями. Простейшее функциональное преобразование входной информации X в выходную информацию Y с помощью функции f имеет следующий вид:

$$Y = f(X) \quad (1)$$

или в форме описания сложной системы, даваемой Месаровичем [6]

$$A_1 : X_1 \rightarrow Y_1. \quad (2)$$

Выражение (2) интерпретируется так: алгоритм (процесс) A_1 осуществляет функциональное преобразование над входным множеством X_1 и преобразует его в промежуточное множество Y_1 . Выражений типа (2) может быть достаточно много для разных последовательных алгоритмических преобразований $A_1, A_2, A_3, \dots, A_i, \dots, A_n$. Все они могут быть обобщенно описаны выражением

$$A_S : X \rightarrow Y. \quad (3)$$

Здесь $A_S = f_A(A_1, A_2, A_3, \dots, A_i, \dots, A_n)$ — составной алгоритм. Выражение (3) интерпретируется так: составной алгоритм A_S осуществляет функциональное преобразование над входным множеством X и преобразует его в выходное множество Y . Если функция f_A — линейная, то алгоритм A_S является линейным. Независимо от вида функции f_A выражение (3) описывает прямой алгоритм.

Условно назовем этот алгоритм "номер один" Al_1 : для него существует описание

$$\exists Al_1(X) \rightarrow \{f(X) \rightarrow Y\}. \quad (4)$$

На рис. 1 приведены модифицированные схемы алгоритмов по отношению к (3). Алгоритм на схеме рис. 1, а называют циклическим. Его описание имеет следующий вид:

$$Al_2 : F((X, n, \delta)) \rightarrow Y(n, \delta), \quad (5)$$

где n — число циклов; δ — погрешность. Выражение (5) означает, что выходное множество $Y(n, \delta)$ получается с применением циклических вычислений и содержит погрешность δ . Недостатком циклического алгоритма является возможность закливания или зависания. Эти ситуации возникают, если алгоритм составлен некорректно или если происходит компьютерный сбой. Алгоритм на рис. 1, а имеет один вход и один выход. Н. Моисеев [1] отмечает, что живые формы имеют обратные связи, без которых невозможно обеспечить сохранение их гомеостазиса и развития. Поэтому циклические алгоритмы важны для моделирования развивающихся и саморазвивающихся систем.

Алгоритм на рис. 1, б называют стратифицированным [7]. Он содержит множество входов и выходов. Его отличие в том, что он создает промежуточные результаты L , которые могут быть использованы независимо или в совокупности. Его описание имеет вид

$$Al_3 = \Sigma Al_i, \quad (6)$$

где $i = 1, \dots, n$ — число слоев L . Выражение (6) означает, что общий алгоритм стратифицированной обработки Al_3 представляет собой сумму частных алгоритмов, каждый из которых имеет свой вход и выход, но в совокупности они обрабатывают общий исходный массив X , который представляет собой сумму непересекающихся множеств. Алгоритм получения отдельного слоя имеет вид

$$Al_i: F_i(X_i) \rightarrow L_i. \quad (7)$$

Выражение (7) означает, что алгоритм обработки отдельного слоя Al_i преобразует входное подмножество X_i в выходное подмножество L_i , которое представляет собой частный результат в общей системе слоев. Система слоев формируется как композиция

$$Y = \Sigma L_i + Con(i, j). \quad (8)$$

Выражение (8) означает, что результат обработки Y представляет собой совокупность слоев L_i и множество связей между ними $Con(i, j)$. Результат стратифицированного алгоритма формируется как сумма результатов, получаемых

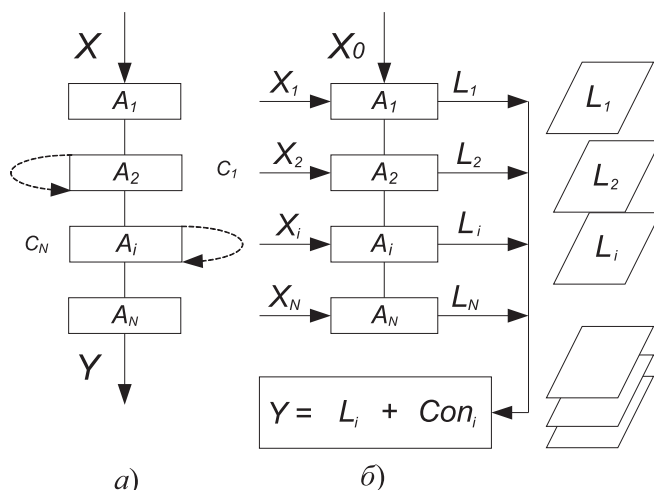


Рис. 1. Циклический (а) и стратифицированный (б) алгоритмы

на каждом слое. При этом сумма результатов может комбинироваться в любой последовательности и любой комбинации. Механизм стратификации применяют в геоинформационных системах [8] и в системах электронной картографии. Стратифицированный алгоритм в отличие от двух предыдущих алгоритмов формирует не окончательный результат, а набор стратов для получения нужного результата путем любой комбинации из данного набора. Стратификацию и стратифицированный алгоритм применяют при выполнении условий

$$X_i \cap X_k = \emptyset; Y_i \cap Y_k = \emptyset, \quad (9)$$

где $i \neq k$,

$$X_i \otimes X_k; Y_i \otimes Y_k, \quad (10)$$

где $\otimes$ — произведение Декарта, $i \neq k$.

Альтернативой стратифицированному алгоритму является итеративный алгоритм [9]. Он имеет схожую структуру, но иной механизм, который формирует окончательный единственный результат. Он не требует выполнения условий (9, 10). На рис. 2 приведены схемы итеративного (рис. 2, а) и линейного инкрементного (рис. 2, б) алгоритмов. При их использовании задается некое начальное условие X_0 , базовое входное множество X_1 и совокупность входных множеств X_i . Это первое принципиальное отличие. В линейных алгоритмах входное множество одно. Итеративный алгоритм (рис. 2, а) имеет следующее описание:

$$Al_4 = \Sigma Al_i, \quad (11)$$

где $i = 1, \dots, n$ — число итераций;

$$Al_i: F_i(X_i) \rightarrow Y_i + d_i. \quad (12)$$

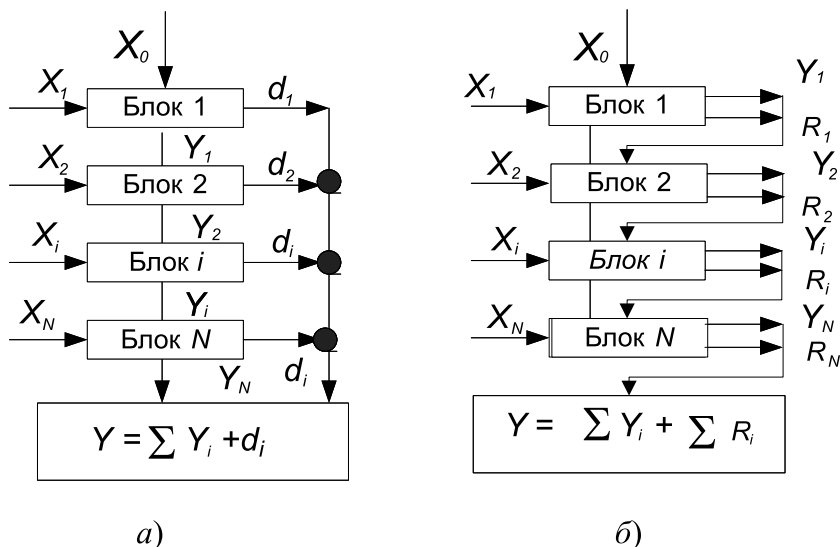


Рис. 2. Итеративный (а) и инкрементный алгоритмы (б)

Выражение (12) означает, что частный алгоритм обработки Al_i преобразует входное подмножество X_i в выходное подмножество Y_i , плюс итерацию d_i , которое представляет собой аддитивную часть в общем выходном множестве Y . Вычисления происходят по двум контурам: основному (Y_i) и вспомогательному (d_i). Результат обработки Y представляет собой аддитивную систему выходных множеств Y_i и итеративных добавок. Для итеративных алгоритмов существует проблема сходимости. Она состоит в том, что не при всяких начальных условиях X_0 результат вычислений должен сходиться:

$$\lim_{i \rightarrow \infty} (Y_i) = Y; \lim_{i \rightarrow \infty} (d_i) = 0. \quad (13)$$

Результат итеративного вычисления Y_i стремится к некоторому устойчивому значению, а значение поправки d_i стремится к нулю.

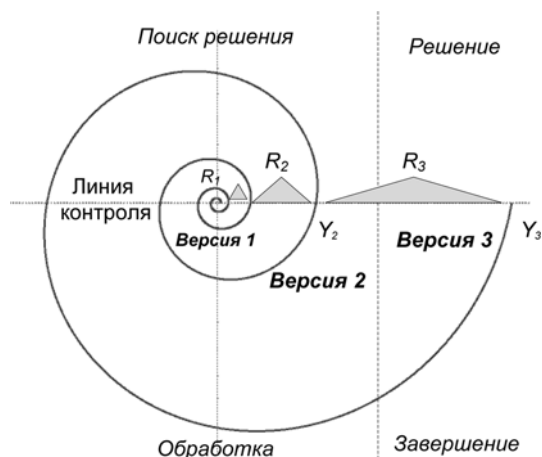


Рис. 3. Спиральная схема инкрементного алгоритма

Инкрементный алгоритм (рис. 2, б) имеет следующее описание [10]:

$$Al_5 = \Sigma Al_i, \quad (14)$$

где $i = 1, \dots, n$ — число инкрементных операций,

$$Al_i: F_i[X_i + R_{(i-1)}] \rightarrow Y_i + R_i. \quad (15)$$

Выражение (14) означает, что общий алгоритм инкрементной обработки Al_5 представляет собой сумму частных алгоритмов, как при стратифицированной (3) и итеративной (6) обработке. Результат инкрементной обработки представляет собой сумму частных решений и сумму полученных

ресурсов, которые не являются решением, но служат инструментом для возможного решения и анализа других задач:

$$Y = \Sigma (Y_i + R_i). \quad (16)$$

Выражение (16) означает, что при инкрементной обработке частный алгоритм обработки Al_i преобразует входное подмножество X_i в выходное подмножество Y_i и создает дополнительный ресурс R_i , который используют на последующем этапе обработки информации. Выражение (16) означает, что результат инкрементной обработки Y представляет собой аддитивное множество Y и плюс дополнительно некий информационный ресурс R , который можно использовать при решении других задач. Алгоритм на рис. 2, б называют линейным в силу того, что он последовательно решает задачу без возврата к предыдущим этапам. В отличие от уменьшающейся поправки при итеративном алгоритме ресурс при инкрементной обработке имеет тенденцию к росту:

$$\lim_{i \rightarrow \infty} (Y_i) = Y; \lim_{i \rightarrow \infty} (R_i) = R_0. \quad (17)$$

Существует модификация инкрементного алгоритма, которую называют "спиральной" или "циклической". Каждый цикл формирует очередную версию решения. Спиральный алгоритм приведен на рис. 3. Схема этого алгоритма показывает, что каждая версия приближает результат к области решения задачи. Решение формируется как функция в полярных координатах $Y = \phi(\varphi)$. Здесь Y — решение или результат обработки; ϕ — радиальная функция; φ — угловой аргумент, характеризующий стадию

(цикл) процесса обработки. При $\varphi = 2\pi$ происходит завершение цикла обработки и формирование очередной версии решения.

Инкрементный спиральный алгоритм имеет описание, аналогичное (14):

$$Al_6 = \Sigma Al_i. \quad (18)$$

Выражение (18) означает, что общий алгоритм обработки Al_6 представляет собой сумму частных алгоритмов, как при линейной инкрементной (14) обработке. Здесь $i = 1, \dots, n$ — число инкрементных операций. Введем понятие инкрементной функции:

$$\begin{aligned} \text{Incf}(Y_i, R_i) = \\ = \text{Incf}(Y_{i-1}, R_{i-1}) + dY_{i-1}(dR_{i-1}). \end{aligned} \quad (19)$$

Конкретный вид функции может быть разным. Смысл формулы (19) в том, что на каждом витке или цикле решения формируется новая версия решения, которая зависит от предыдущей версии и прироста ресурса, используемого на начальной стадии формирования данной версии. Используя системный подход [11, 12], можно описать спиральный инкрементный алгоритм IncA:

$$\begin{aligned} \text{IncA: Incf}(Y_i, R_i) = \\ = \text{Incf}(Y_{i-1}, R_{i-1}) + dY_{i-1}(dR_{i-1}), \end{aligned} \quad (20)$$

где Y_i — частные решения на каждом шаге; $i = 1, \dots, n$, n — число инкрементных спиралей; R_i — частный ресурс. Выражение (20) означает, что инкрементный алгоритм обработки преобразует входное подмножество (версию решения) Y_{i-1} в выходное подмножество Y_i (новую версию) и создает дополнительный ресурс R_i , который используют, если необходимо, в следующем цикле обработки. Дополнительно на каждом витке спирали формируется дополнительное условие решения задачи, которое меняет вектор решения. Обработка осуществляется с возвратом к линии контроля, но с получением дополнительных ресурсов и дополнительных условий.

Итоговое решение получают как

$$Y = \Sigma [Y_i + R_i] + \Sigma C_j, \quad (21)$$

где C_j — дополнительные условия.

Выражение (21) означает, что результат инкрементной обработки Y представляет собой аддитивное множество Y , в которое интегрирован инкрементный ресурс, на-

капливаемый при реализации каждой версии обработки, и дополнительные условия, которые могут дополнять или уточнять первоначальные условия решения задачи.

Таким образом, при инкрементном решении получают три результата: решение задачи (Y); дополнительный ресурс (R); уточненные (или дополненные) исходные условия решения задачи (C). Последнее подразумевает, что инкрементный алгоритм можно применять тогда, когда исходные условия нечетко сформулированы или являются некорректными. Это расширяет сферу применения алгоритма в область некорректно поставленных задач.

Алгоритм на рис. 3 называют спиральным в силу того, что он циклически решает задачу с возвратом к предыдущим этапам, но с наращиванием ресурса решения задачи и изменением (уточнением) исходных условий. Спираль раскручивается до тех пор, пока очередная версия не попадет в область решения. Алгоритм на рис. 3 обобщенно описывает генетические алгоритмы и мультиагентные системы [13]. Схема на рис. 3 является частным случаем фазового портрета линейной однородной системы:

$$x' = ax + by = \bar{\xi}_1; y' = cx + dy = \bar{\xi}_2. \quad (22)$$

Здесь x', y' — производные. Одной из интегрирующих функций для системы (22) является функция

$$M(x, y) = -cx^2 + (a - d)xy + by^2. \quad (23)$$

Она представляет собой квадратичную форму относительно (x, y) . Не вдаваясь в детали, выделим две фазовые характеристики, связанные с векторными полями $\bar{\xi}_1, \bar{\xi}_2$. Они приведены на рис. 4.

Рис. 4, а соответствует схеме рис. 3 и описывает развитие инкрементного процесса, при

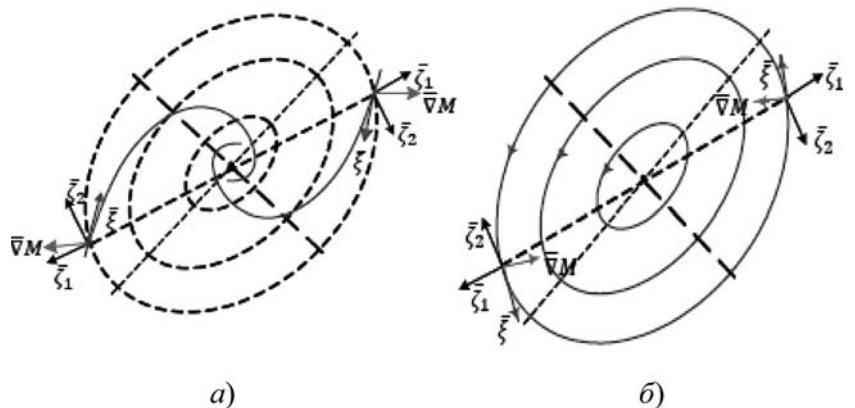


Рис. 4. Варианты развития инкрементного решения

котором рано или поздно спираль попадает в область решений. Рис. 4, б описывает состояние инкрементного процесса, при котором спираль трансформируется в эллипс, процесс решения заклинивается и не попадает в область решений. Такая информационная ситуация обусловлена трансформацией инкрементных условий. Они могут как "разворачивать" спираль, так и "сворачивать" ее. Поэтому не во всех случаях применение инкрементного алгоритма приводит к получению решения.

Сравнение алгоритмов первого и второго рода

Рассмотренные алгоритмы относятся к алгоритмам первого рода. Алгоритм первого рода возможен при выполнении когнитивных условий: обозримости, воспринимаемости и интерпретируемости условий задачи. Он приведен на рис. 5.

В этом случае человек в силу своего естественного интеллекта способен воспринять информационную ситуацию, которая обладает свойствами обозримости, воспринимаемости и интерпретируемости. В этом случае субъект может применить один из алгоритмов, рассмотренных выше: $Al_1, Al_2, Al_3, Al_4, Al_5, Al_6$. Возможна ситуация, при которой возникают оппозиционные условия: "необозримость", "невоспринимаемость" [14, 15], "неинтерпретируемость". В этом случае ясна начальная ситуация, но неясен путь достижения цели и возникает сложность в выборе маршрута достижения цели. Важным фактором для алгоритма является то, что его можно рассматривать как систему, которая имеет собственные цели и располагает определенными ресурсами для их достижения. Если ресурсов недостаточно, то возникает задача второго рода [16] и алгоритм второго рода [17]. Это показано на рис. 6. Задачи второго рода характеризуются неоднозначностью решения и отсутствием единственного алгоритма, который приводит от начальной ситуации к конечной точке. Граф задачи второго рода имеет циклы. На начальной стадии возможны промежуточные решения Z .

В результате решения задачи второго рода возможны варианты решения. Решение может быть оптимальным Y_o . Решение может быть не оптимальным Y_{NO1} . Решение может быть ошибочным Y_{false} . Это обусловлено частичной воспринимаемостью условий задачи и возможными путями решения. В этом случае применяют эвристические методы поиска решений, в чис-

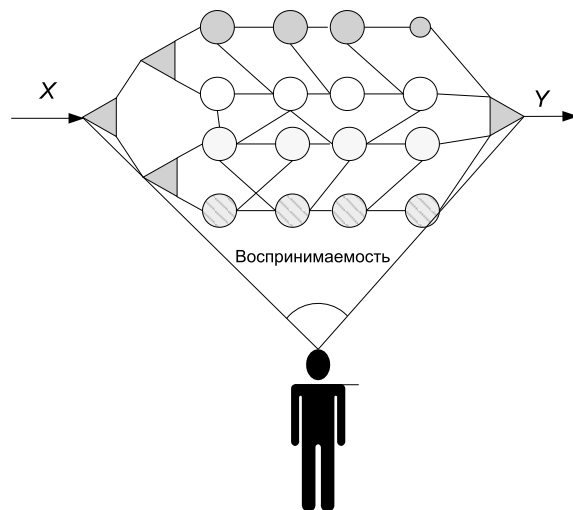


Рис. 5. Алгоритм первого рода

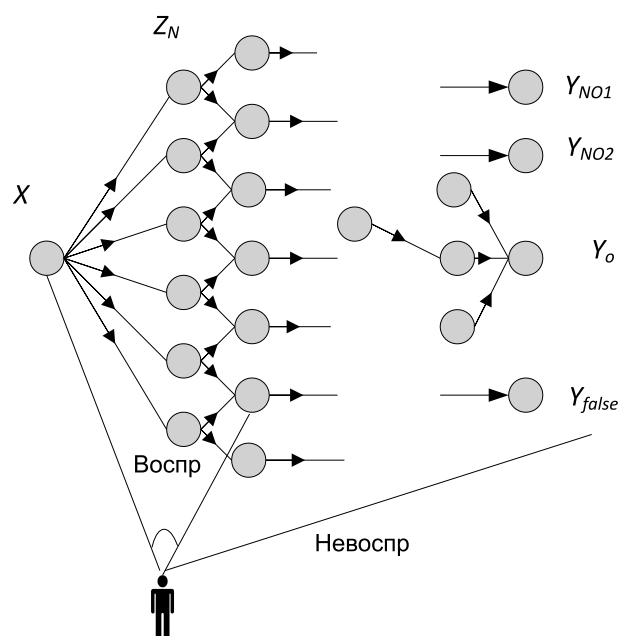


Рис. 6. Информационная ситуация возникновения алгоритма второго рода

ло которых входит также инкрементный алгоритм. Для задач первого рода инкрементный алгоритм характерен схемой на рис. 3, для задач второго рода инкрементный алгоритм имеет варианты на рис. 4.

Для формирования алгоритмов второго рода применяют кроме инкрементного подхода систему правил поиска, включая игровые методы, например метод Монте Карло [18]. Широко распространенным алгоритмом второго рода является муравьиный алгоритм [19]. Муравьиный алгоритм моделирует алгоритм оптимизации поведения муравьиной колонии (*ant colony optimization*) при поиске пищи. Он используется для нахождения приближенных решений транс-

портной задачи, а также для поиска маршрутов на графах. Отдаленно он напоминает "метод Монте Карло с условием". Алгоритм заключается в использовании системы правил модели группового поведения. В версии алгоритма Марко Дориго [20] осуществлялся поиск оптимального пути в графе на основе двух групп правил. Первая группа использовала модель "жадного алгоритма" [21], вторая группа дополняла его правилами "стадности" [22]. Правила "жадность" характеризуют локальную мотивацию, правила "стадность" характеризуют глобальную мотивацию и исключают "случайность". Передача информации создает возможность организации колонии муравьев в достижении цели. Такая модель используется в мультиагентных системах. В алгоритмах второго рода, основанных на множестве правил коллективного поведения, обратные связи возникают на основе установившихся (применяемых) правил коллективного поведения. Мультиагентная система [11] ищет оптимальное решение только на основе системы правил, находя его под конкретные начальные условия. В этом случае оптимальным решением является такое решение, на которое потрачено наименьшее количество ресурсов. Это опять связывает такой подход с инкрементным алгоритмом, в котором существует параметр "ресурсы".

Заключение

Проведенный анализ дает основание утверждать, что современное расширенное понятие алгоритма охватывает не только область вычислений, но и область рассуждений, и область решения сложных проблем. Расширенное понимание алгоритма связано в первую очередь с алгоритмами второго рода, которые представляют собой поэтапные методы решения сложных задач или проблем. Алгоритм первого рода или прямое решение позволяет исключить участие человека в процессе решения или рассуждения. Этим повышается скорость вычислений. Алгоритм второго рода или поэтапное решение проблемы требует включения человека или систем искусственного интеллекта в решение задачи или проблемы. Обобщение применения алгоритмов первого и второго рода позволяет получать новое знание.

Расширенное понимание алгоритма дает возможность привлечения методов построения алгоритмов из разных областей: биологии, генетики, искусственного интеллекта. Принципи-

альным является использование когнитивных характеристик воспринимаемости и обозримости при начальном формировании алгоритмов. При наличии воспринимаемости и обозримости существует возможность построения алгоритмов первого рода. При невоспринимаемости и необозримости требуется использовать алгоритмы второго рода. При недостаточности ресурсов для достижения цели также необходимо применять алгоритмы второго рода. Некоторые виды алгоритмов содержат познавательные функции. Стратифицированные алгоритмы и спиральные (инкрементные) алгоритмы можно использовать для решения задач второго рода с некоторыми условностями. Стратифицированный алгоритм приводит только к разбиению начального множества без установления полной системы связей между слоями. Современное развитие методов вычислений и алгоритмов требует исследований алгоритмов второго рода. Важной поддержкой этого направления может служить системная инженерия. Перспективным следует считать развитие методов когнитивного формирования алгоритмов. Онтологический инжиниринг также может служить основой современного построения алгоритмов первого и второго рода.

Список литературы

1. Моисеев Н. Алгоритмы развития. М.: Наука, 1987. 304 с. (Переиздана в 2017 издательством Litres).
2. Nonaka I. A dynamic theory of organizational knowledge creation // Organization Science. 1994. Vol. 1. N. 5. P. 14—37.
3. Sigov A. S., Tsvetkov V. Ya. Tacit Knowledge: Oppositional Logical Analysis and Typologization // Herald of the Russian Academy of Sciences. 2015. Vol. 85, N. 5. P. 429—433. DOI: 10.1134/S1019331615040073.
4. Cormen T. H., Leiserson Ch. E., Rivest R. L. Introduction to Algorithms. 1st. MIT Press and McGraw-Hill, 1990. 863 p.
5. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ: пер. с англ. М.: Вильямс, 2009. 1296 с.
6. Месарович М., Такахара Н. Общая теория систем: математические основы. М.: Мир, 1978. 311 с.
7. Буравцев А. В. Стратифицированный метод построения сложной системы // Образовательные ресурсы и технологии. 2017. № 3 (20). С. 23—32.
8. Цветков В. Я. Геоинформационные системы и технологии. М.: Финансы и статистика, 1998. 288 с.
9. Wang Z., Han B. Advanced iterative algorithm for phase extraction of randomly phase-shifted interferograms // Optics letters. 2004. Vol. 29. N. 14. P. 1671—1673.
10. Цветков В. Я., Железняков В. А. Инкрементальный метод проектирования электронных карт // Инженерные изыскания. 2011. № 1. С. 66—68.
11. Кудж С. А. Системный подход // Славянский форум. 2014. № 1 (5). С. 252—257.
12. Цветков В. Я. Решение проблем с использованием системного анализа // Перспективы науки и образования. 2015. № 1. С. 50—55.

13. Розенберг И. Н., Цветков В. Я. Применение мультиагентных систем в интеллектуальных логистических системах // Международный журнал экспериментального образования. 2012. № 6. С. 107—109.

14. Буравцев А. В. Серый управленческий анализ // Перспективы науки и образования. 2017. № 5 (29). С. 74—79.

15. Цветков В. Я. Когнитивные аспекты построения виртуальных образовательных моделей // Перспективы науки и образования. 2013. № 3. С. 38—46.

16. Tsvetkov V. Ya. Incremental Solution of the Second Kind Problem on the Example of Living System // Biosciences biotechnology research Asia. November 2014. Vol. 11 (Spl. Edn.). P. 177—180.

17. Doha E. H., Abd-Elhameed W. M., Youssri Y. H. Second kind Chebyshev operational matrix algorithm for solving differential

equations of Lane—Emden type // New Astronomy. 2013. Vol. 23. P. 113—117.

18. Robert C. P. Monte Carlo methods. John Wiley & Sons, Ltd, 2004.

19. Bautista J., Pereira J. Ant algorithms for assembly line balancing // Ant algorithms. 2002. P. 49—61.

20. Dorigo M., Di Caro G., Gambardella L. M. Ant algorithms for discrete optimization // Artificial life. 1999. Vol. 5. N. 2. P. 137—172.

21. DeVore R. A., Temlyakov V. N. Some remarks on greedy algorithms // Advances in computational Mathematics. 1996. Vol. 5. N. 1. P. 173—187.

22. Wang Gai-Ge, Guo Lihong, Gandomi Amir H. et al. Chaotic krill herd algorithm // Information Sciences. 2014. N. 274. P. 17—34.

V. Ya. Tsvetkov, D. Sc., Professor, Deputy Head,

Center Fundamental and Advanced Research, Research and Design Institute of Design Information,
Automation and Communication on Railway Transport, SC NIIAS,
Moscow, 109029, Russia, e-mail: cvj2@mail.ru

The Algorithms as a Means of Knowledge

The article explores algorithms as a means of cognition. It is shown that the algorithm is not only a means of calculation, but a means of storing knowledge. Algorithm is also a means of transferring knowledge. The article claims that the algorithm is a means of transforming knowledge. The connection between the algorithm and the model of transformation of implicit knowledge is shown. The article claims that many complex algorithms can be considered as complex systems. This makes it possible to apply system analysis to the analysis of algorithms.

The paper describes two classes of problems and two classes of algorithms based on the receipt of a solution. Algorithms of the first kind and algorithms of the second kind. Algorithms of the first kind allow one to obtain a direct solution of the problem in one step. Algorithms of the second kind solve the problem stage after stage. The article shows the difference between linear and nonlinear algorithms. The article introduces a formal description of algorithms based on system analysis. The article analyzes six typical algorithms of the first kind. An evolutionary relationship between algorithms of the first kind is shown. The article notes the similarity and difference between these algorithms. The article gives a criterion for assigning an algorithm to the first or second kind. The article contains structural diagrams of typical algorithms of the first kind. For each structural scheme, a formal description of the algorithm is given. The difference between the iterative algorithm and the incremental algorithm is shown. The article analyzes the reasons for the emergence of an algorithm of the second kind. It is shown that the main criterion for the appearance of algorithms of the second kind are cognitive signs of foreseeability and perceptibility. The article analyzes algorithms of the second kind and determines the methods of their construction. The methods of solution for the occurrence of problems of the second kind are described. The possibility of using multi-agent systems for solving problems of the second kind.

Keywords: calculation, cognition, algorithm, linear algorithm, direct algorithm, algorithm of the first kind, algorithm of the second kind, perceivability of the problem

DOI: 10.17587/it.24.507-515

References

1. Moiseev N. Algoritmy razvitiya, Moscow, Nauka, 1987, 304 p. Pereizdana 2017, Izdatel'stvom Litres (in Russian).

2. Nonaka I. A dynamic theory of organizational knowledge creation. *Organization Science*, 1994, no. 5 (1), pp. 14—37.

3. Sigov A. S., Tsvetkov V. Ya. Tacit Knowledge: Oppositional Logical Analysis and Typologization, *Herald of the Russian Academy of Sciences*, 2015, vol. 85, no. 5, pp. 429—433. DOI: 10.1134/S1019331615040073.

4. Cormen T. H., Leiserson C. E., Rivest R. L. Introduction to Algorithms. 1st. MIT Press and McGraw-Hill, 1990. 863 p.

5. Kormen T., Leiserson Ch., Rivest R. et al. Algoritmy. Postroenie i analiz: [per. s angl.], Moscow, Vil'jams, 2009. 1296 p. (in Russian).

6. Mesarovich M., Takahara N. Obshhaja teoriya sistem: matematicheskie osnovy, Moscow, Mir, 1978, 311 p. (in Russian).

7. Buravcev A. V. Stratificirovannyj metod postroenija slozhnoj sistemy, *Obrazovatel'nye resursy i tehnologii*, 2017, no. 3 (20), pp. 23—32 (in Russian).

8. Tsvetkov V. Ya. Geoinformacionnye sistemy i tehnologii, Moscow, Finansy i statistika, 1998, 288 p. (in Russian).

9. Wang Z., Han B. Advanced iterative algorithm for phase extraction of randomly phase-shifted interferograms, *Optics letters*, 2004, vol. 29, no. 14, pp. 1671—1673.

10. Tsvetkov V. Ya., Zheleznyakov V. A. Inkremental'nyj metod proektirovaniya jelektronnyh kart, *Inzhenernye izyskaniya*, 2011, no. 1, pp. 66—68 (in Russian).

11. Kudzh S. A. Sistemnyj podhod, *Slavjanskij forum*, 2014, no. 1 (5), pp. 252—257 (in Russian).

12. Tsvetkov V. Ya. Reshenie problem s ispol'zovaniem sistemnogo analiza, *Perspektivy nauki i obrazovaniya*, 2015, no. 1, pp. 50—55 (in Russian).

13. Rozenberg I. N., Tsvetkov V. Ya. Primenenie mul'tiagentnyh sistem v intellektual'nyh logisticheskikh sistemah, *Mezhdunarodnyj*

zhurnal jeksperimental'nogo obrazovaniya, 2012, no. 6, pp. 107—109 (in Russian).

14. **Buravcev A. V.** Seryj upravlencheskij analiz, *Perspektivy nauki i obrazovaniya*, 2017, no. 5 (29), pp. 74—79 (in Russian).

15. **Tsvetkov V. Ya.** Kognitivnye aspekty postroeniya virtual'nyh obrazovatel'nyh modelej, *Perspektivy nauki i obrazovaniya*, 2013, no. 3, pp. 38—46 (in Russian).

16. **Tsvetkov V. Ya.** Incremental Solution of the Second Kind Problem on the Example of Living System, *Biosciences biotechnology research Asia*, November 2014, vol. 11 (Spl. Edn.), pp. 177—180.

17. **Doha E. H., Abd-Elhameed W. M., Youssri Y. H.** Second kind Chebyshev operational matrix algorithm for solving differential equations of Lane—Emden type, *New Astronomy*, 2013, vol. 23, pp. 113—117.

18. **Robert C. P.** Monte Carlo methods, John Wiley & Sons, Ltd, 2004.

19. **Bautista J., Pereira J.** Ant algorithms for assembly line balancing, *Ant algorithms*, 2002, pp. 49—61.

20. **Dorigo M., Di Caro G., Gambardella L. M.** Ant algorithms for discrete optimization, *Artificial life*, 1999, vol. 5, no. 2, pp. 137—172.

21. **DeVore R. A., Temlyakov V. N.** Some remarks on greedy algorithms, *Advances in computational Mathematics*, 1996, vol. 5, no. 1, pp. 173—187.

22. **Wang Gai-Ge, Guo Lihong, Gandomi Amir H.** et al. Chaotic krill herd algorithm, *Information Sciences*, 2014, vol. 274, pp. 17—34.

УДК 004.912

DOI: 10.17587/it.24.515-523

Ю. С. Шучалова, приглашенный преподаватель, iusshuchalova@hse.ru,

В. В. Ланин, ст. преподаватель, vlanin@hse.ru,

Национальный исследовательский университет "Высшая школа экономики", г. Пермь

Исследовательский портал для анализа и оценки стиля научных публикаций

Описан этап проектирования портала для проведения корпусных исследований английского языка. Сформулированы требования к решению, показаны лингвистические подходы к решению поставленных задач. Приведен процесс моделирования системы и рассмотрены особенности реализации с учетом специфики предметной области. Для интеграции гетерогенных компонентов предложена сервисная архитектура.

Ключевые слова: корпусные исследования, исследовательский портал, академический английский язык, моделирование информационных систем

Введение

Широкое распространение задач и инструментов, связанных с информационными технологиями, в науках о естественных языках позволило лингвистам наряду с традиционными методами развивать такое направление, как корпусная лингвистика. Данное направление основано на использовании так называемых корпусов, под которыми понимается представительная коллекция текстов по определенной тематике, доступная в электронной форме [1], т. е. набор документов или записей, собранных по определенному принципу: общий стиль, общий автор, общий жанр и т. д. Таким образом, в отличие от классической лингвистики, корпусная лингвистика ориентирована на изучение практического применения языка, а не его теоретических аспектов. Многие традиционные задачи языкознания решаются с помощью текстовых корпусов. Например, изучение отдельных словоупотреблений, общей динамики языка, машинный перевод и

обучение языку (*computer-assisted language learning*), некоторые виды анализа текста. В целом поиск новых возможностей применения корпусов в различных областях — одна из основных задач корпусной лингвистики.

Корпусная лингвистика [2] тесно связана с информационными технологиями. Сбор и обработка обширных корпусов требуют использования программного обеспечения и технологий, допускающих совместную работу множества удаленных пользователей. Программное обеспечение, предназначенное для работы лингвистов, как правило, можно разделить на общее — для выполнения базовых функций (например, автоматической морфологической разметки) — и специализированное, созданное для решения конкретных задач. Одной из таких задач является проверка соответствия текста академическому стилю. При написании научных статей на английском языке студенты сталкиваются с необходимостью использования стандартных языковых средств для соблюдения академической стилистики текста.