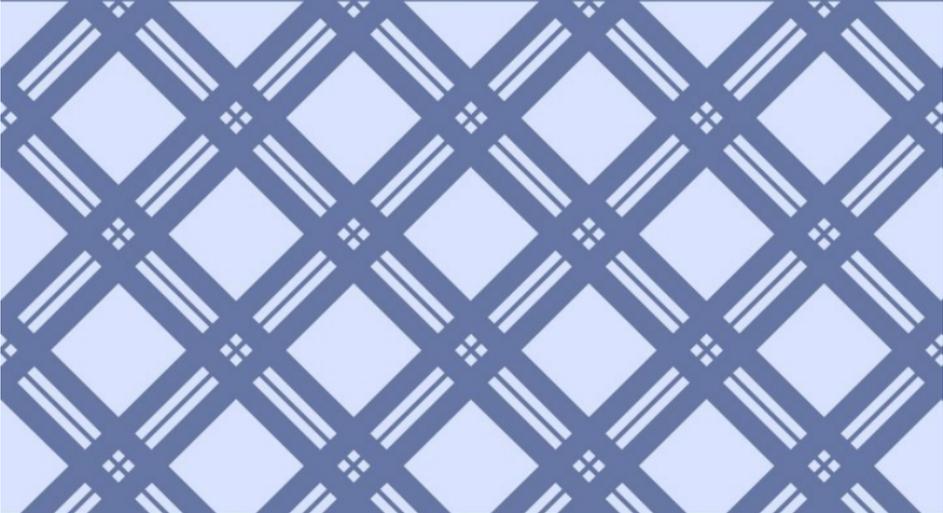


Геннадий Степанов



**ВЫЧИСЛИТЕЛЬНЫЕ
МАШИНЫ
И ТРУДНОРЕШАЕМЫЕ
ЗАДАЧИ**

Русский метод. Русская машина

Геннадий Степанов
Вычислительные машины
и труднорешаемые задачи.
Русский метод. Русская машина

http://www.litres.ru/pages/biblio_book/?art=63470768
ISBN 9785005189608

Аннотация

В этой, предлагаемой мной умному, любознательному и доброжелательно настроенному читателю, книге, описываются некоторые примеры решения труднорешаемых задач. В этих примерах показываются возможные, в общем виде, некоторые приёмы применения Русского метода при решении NP-задач. Таких приёмов (вариантов) применения Русского метода может быть неограниченное множество для получения как приближённых, так и оптимальных решений NP-задач без зацикливания.

Содержание

ВВЕДЕНИЕ	5
Задача о вершинном покрытии	9
Задача о доминирующем множестве	14
Конец ознакомительного фрагмента.	15

**Вычислительные машины
и труднорешаемые задачи
Русский метод.
Русская машина**

Геннадий Степанов

© Геннадий Степанов, 2020

ISBN 978-5-0051-8960-8

Создано в интеллектуальной издательской системе Ridero

Вычислительные машины и труднорешаемые зада-

чи

Русский метод Русская машина

ВВЕДЕНИЕ

В этой, предлагаемой мной умному, любознательному и доброжелательно настроенному читателю, книге, описываются **некоторые примеры** решения труднорешаемых задач.

В этих примерах показываются возможные, в общем виде, **некоторые приёмы** применения Русского метода при решении NP – задач.

Таких приёмов (вариантов) применения Русского метода может быть неограниченное множество для получения, как приближённых, так и оптимальных решений NP – задач без зацикливания.

Русский метод решения NP – задач не отличается особой сложностью. По моему мнению, его легко может понять, на интуитивном уровне, любой школьник.

Существующие методы решения NP – задач являются методами неявного полного перебора. Перебор – это составная часть сортировки.

Русский метод использует сортировку, ограниченную числом угадывания (N уг). Сортировка применяется в Русском методе на каждом промежуточном этапе решения NP – задач.

Отличительной особенностью Русского метода является то, что он основан на найденной мной закономерной зависи-

мости, присущей NP – задачам.

Главным достоинством этой зависимости является возможность создать имитатор человеческого разума и мышления на основе философии Великого философа Иммануила Канта.

В основе данного имитатора лежит разработанный мной механизм машинной интуиции.

По Канту сущность вещей в мире не познаваема человеком. Они ноумены. Человек способен познать только явления в мире.

Невозможно познать человеческий разум и мышления по Канту. Они ноумены.

Хотя создать имитатор человеческого разума и мышления, основанного на выявленных наукой явлениях, вполне возможно, по Канту.

Русский метод, не отличаясь особой теоретической сложностью, требует по своей методологии совершенно другие вычислительные устройства, в отличие от существующих современных вычислительных машин

Русский метод очень сложно приспособить для решения задач на этих вычислительных устройствах, так как они основаны на **классической фон-неймановской логике**, — так называемое **бутылочное горлышко архитектуры фон Неймана (von Neumann bottleneck)**.

Для решения задач с помощью Русского метода необходимо вычислительное устройство с практически с неограни-

ченным числом параллельных вычислительных путей, что невозможно осуществить на современных вычислительных машинах, основанных на архитектуре фон Неймана, которым требуется для вычислений колоссальные энергозатраты.

Мной предлагается для решения задач **Русским методом** совершенно другое вычислительное устройство, а именно **Русскую машину**.

Архитектура Русской машины **обходит серьёзное препятствие в классической фон-неймановской логике**, — так называемое **бутылочное горлышко архитектуры фон Неймана (von Neumann bottleneck)**.

Это ограничение в современных вычислительных устройствах связано с необходимостью извлекать огромные массивы данных из памяти, которые затем пересылаются для обработки в процессор.

Время извлечения данных и их пересылка могут оказаться много больше времени, необходимого на их обработку процессором современных вычислительных устройств.

Это особенно критично для работы ускорителей нейронных сетей, которые опираются на операции с перемножением массивных векторных матриц, а это всё затраты энергии и немалой.

Архитектура Русской машины делает всё иначе.

Русская машина производит вычисления непосредственно в памяти аналогового мультипроцессора и делает это не с цифровыми данными, а с данными, представленными

в аналоговом виде.

«Аналоговые» технологии позволяют получить практически тот же результат при перемножении векторных матриц с допущением меньшей точности, чем при использовании данных в виде цифровых.

Экономия происходит сразу по двум пунктам:

- по пересылке данных из памяти в аналоговый мультипроцессор;
- по объёму используемых для расчётов данных, практически неограниченных.

Задача о вершинном покрытии

Задача о вершинном покрытии – NP-полная задача информатики в области теории графов.

Часто используется в теории сложности для доказательства NP-полноты более сложных задач.

Вершинное покрытие для неориентированного графа $G = (V, E)$ – это множество его вершин S , такое, что, у каждого ребра графа хотя бы один из концов входит в вершину из S .

Размером (size) вершинного покрытия называется **число входящих в него вершин**.

Задача о вершинном покрытии состоит в поиске вершинного покрытия **наименьшего размера** для заданного графа (этот размер называется числом вершинного покрытия графа).

NP-полнота задачи о вершинном покрытии.

Поскольку задача о вершинном покрытии является NP-полной, то, в настоящее время считается, что неизвестны алгоритмы для её решения за полиномиальное время.

Однако существуют алгоритмы, дающие «**приближённое**» решение этой задачи за полиномиальное время.

С их помощью можно найти, например, вершинное покрытие, в котором число вершин не более чем вдвое превосходит минимально возможное.

Безпереборный метод (русский метод) решения за-

дачи о вершинном покрытии графа.

Мной предлагается следующий способ решения задачи о вершинном покрытии:

Этап1.

Выберем интервал изменения числа угадывания (N уг) для данной конкретной задачи. Этот интервал зависит от характеристик этой задачи.

Он изменяется от 1 до определённого, предварительно, для этой задачи значения (N макс).

Процесс изменения можно осуществлять, например, путём прибавления 1 к значению N уг.

Это не является принципиально важным.

Этап2.

Осуществим сортировку вершин графа, в соответствии с числом входов в вершину графа рёбер графа.

Этап3.

Выберем из отсортированного множества вершин графа N уг «лучших» вершин (больше количество входов рёбер графа в вершину), в соответствии с числом входов в вершину графа рёбер графа.

Этап4.

Объединим каждую вершину графа с другой каждой вершиной графа во множества по две вершины.

Определим число входов в вершину графа рёбер графа для данного множества по два, которые запомним для этого множества.

Получим множество подмножеств по два.

Определим, получен ли искомый результат или нет.

Если получен, то заканчиваем поиск.

Переход к этапу 12.

Иначе переходим к следующему этапу.

Этап 5.

Осуществим сортировку подмножеств по два, в соответствии с числом входов в вершину графа рёбер графа.

Этап 6.

Выберем из отсортированного множества подмножеств по два графа N уг лучших подмножеств, в соответствии с числом входов в вершину графа для этих подмножеств.

Этап 7.

Объединим каждое подмножество по два в подмножества по три или четыре в зависимости от условий задачи.

Так, как осуществляется склейка путей графа русским методом в задаче коммивояжера.

Определим число входов в вершину графа рёбер графа для данного множества вершин по три или четыре, которые запомним для этого подмножества.

Получим множество подмножеств по три или четыре.

Если множество подмножеств по три или по четыре оказывается пустым, то увеличиваем N уг, допустим на 1, и переходим к **этапу 3**.

Определим, получен ли искомый результат или нет.

Если получен, то заканчиваем поиск.

Переход к этапу12.

Иначе переходим к следующему этапу.

Этап8.

Осуществим сортировку подмножеств вершин по три или четыре, в соответствии с числом входов в вершину графа рёбер графа, русского метода.

Этап9.

Выберем из отсортированного множества подмножеств вершин по три или четыре N уг лучших подмножеств, в соответствии с числом входов в вершину графа для этих подмножеств.

Этап10.

Объединим каждое подмножество вершин по три в подмножество вершин по шесть или пять.

Или подмножество вершин по четыре во множества подмножеств по восемь или семь.

В полном соответствии с правилами русского метода, указанными в задаче коммивояжера.

Определим число входов в вершину графа рёбер графа для данного множества подмножеств по пять, или шесть, или семь, или восемь, в зависимости от условий задачи, которые запомним для этого множества подмножеств.

Определим, получен ли искомый результат или нет.

Если получен, то заканчиваем поиск.

Переход к этапу12.

Иначе переходим к следующему этапу.

Этап11.

Данную процедуру повторяем, пока число вершин в подмножествах не сравняются с числом вершин графа. Так, как осуществляется эта процедура, примерно, в задаче коммивояжера русским методом.

Определим

$N_{уг} = N_{\text{макс}}$.

Если равен то переходим к этапу 12.

Иначе увеличиваем $N_{уг}$, допустим, на 1 и переходим к этапу 3.

Этап12.

Анализ полученного результата.

Оценка полученного решения.

Если не удовлетворяет то уточняем $N_{уг}$ и $N_{\text{макс}}$.

Переходим к этапу 2.

Иначе заканчиваем работу.

Задача о доминирующем множестве

В теории графов **доминирующее множество** для графа $G = (V, E)$ – это подмножество D множества вершин V , такое, что любая вершина не из D смежна хотя бы одному элементу из D .

Число доминирования $\gamma(G)$ – это число вершин в наименьшем доминирующем множестве G .

Задача о доминирующем множестве заключается в проверке, верно ли неравенство $\gamma(G) \leq K$ для заданного графа G и числа K .

Задача является классической NP-полной проблемой разрешимости в теории вычислительной сложности.

Таким образом, в настоящее время полагают, что **не существует эффективного алгоритма** для нахождения наименьшего доминирующего множества для заданного графа.

Точные алгоритмы

Минимальное доминирующее множество графа с n вершинами может быть найдено за время $O(2^n)$ путём просмотра всех подмножеств вершин.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.