Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра Математического и компьютерного моделирования

Программирование оптимальных транспортных потоков

с помощью эвристических алгоритмов

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы

направление 38.03.05 — Бизнес информатика

механико-математического факультета

Саюрова Никиты Дмитриевича

Научный руководитель
доцент, к.ф. – м.н., доцент ________________ С.П. Шевырев

Зав. кафедрой
зав.каф.,д.ф. – м. н. ________________ Ю.А. Блинков

Саратов 2017

ВВЕДЕНИЕ

Одним из важнейших разделов прикладной математики является теория графов, которая в настоящее время широко используется в самых различных направлениях деятельности человека. Это проблемы построения систем связи, исследование процессов передачи информации, методы построения электрических сетей и многое другое.

Имея в своей основе простейшие элементы – точки, соединенные линиями, – теория графов строит из них богатое многообразие форм, наделяет их интересными свойствами и в результате становится полезным инструментом при исследовании самых разнообразных систем.

Начало этой науке положено выдающимся ученым Леонардом Эйлером при решении знаменитой задачи о «кенигсбергских мостах» в 1736 г. Позднее, в 19 веке, теория графов была применена инженером-электриком Кирхгофом при исследовании электрических цепей, Кэли – в органической химии, Гамильтоном – в картографии, для решения головоломок.

В XX веке теория графов широко использовалась в социально-экономических и биологических науках для исследования проблем экологииохраны окружающей среды. Она также применялась в больших городах при составлении маршрутов движения городских уборочных машин и в других бытовых случаях.

В настоящее время в рамках данной теории также рассматривается большое количество практических задач, которые можно сформулировать и решить как сетевые модели. Приведем несколько конкретных примеров:

1. Проектирование газопровода, соединяющего буровые скважины морского базирования с находящейся на берегу приемной станцией. Целевая функция соответствующей модели должна минимизировать стоимость строительства газопровода.

2. Нахождение кротчайшего маршрута между двумя городами по существующей сети дорог.

3. Определение схемы транспортировки нефти от пунктов Нефтедобытчик к нефтеперерабатывающим заводам

4. Составление временного графика строительных работ (определение дат начала и завершения отдельных этапов работ).

Решение приведенных задач (как и многих других подобных задач) требует применения различных сетевых алгоритмов:

1. Алгоритм нахождения минимального остовного дерева;
2. Алгоритм нахождения кротчайшего пути;
3. Алгоритм определения максимального потока;
4. Алгоритм минимизации стоимости потока в сети с ограниченной пропускной способностью;
5. Алгоритм нахождения критического пути.

Задачи, вытекающие из перечисленных примеров, можно сформулировать и решать как задачи линейного программирования. Однако специфическая структура этих задач позволяет разработать специальные более эффективные сетевые алгоритмы. Рассмотрим наиболее интересные для нас задачи оптимизации потока.

Цель работы – сравнить классические алгоритмы поиска максимального потока в транспортной сети с некоторыми эвристическими алгоритмами, реализующими ту же задачу.

Основное содержание работы

Графы, орграфы и сети — это всего лишь математические абстракции, однако они приносят практическую пользу, поскольку позволяют решать множество важных задач. В этой главе мы расширим сетевую модель решения задач с таким расчетом, чтобы она охватывала динамические ситуации, отождествляющие в нашем воображении движение материалов по сети, в которой различным маршрутам назначаются различные стоимости. Такие расширения позволяют решать удивительно широкие классы задач с длинным списком.

Мы увидим, что эти задачи и приложения могут быть решены с помощью нескольких естественных моделей, при этом мы можем переходить от одной из них к другой посредством метода сведения к другим задачам (reduction). Существует несколько различных технически эквивалентных способов формулирования базовых задач. Для реализации алгоритмов, которые их решают, мы выбираем две конкретные задачи, составляем эффективные алгоритмы их решения, а затем разрабатываем алгоритмы, которые решают другие задачи, отыскивая возможности сведения их к уже известным задачам.

В реальной жизни мы не всегда располагаем свободой выбора, какую предполагает этот идеализированный сценарий, поскольку возможность сведения решения одной задачи к решению другой установлена не для каждой пары задач и поскольку известны лишь немногие оптимальные алгоритмы решения любой из этих задач. Возможно, что пока еще не найдено прямого решения заданной задачи, и вполне вероятно, что пока еще не удалось найти эффективного метода, позволяющего сводить решение одной задачи из заданной пары задач к решению другой.

Приводимый ниже пример служит иллюстрацией широты круга задач, которые могут решаться с помощью моделей потоков в сетях, предлагаемых ею алгоритмов и реализаций. Его можно разбить на общие категории, известные как задачи распределения (distribution), сочетания (matching) и поиска сечения (cut); мы поочередно рассмотрим каждую из них. Мы не будем сосредоточивать внимание на конкретных деталях этих примером, зато выделим несколько различных связанных между собой задач.

Алгоритм *Форда–Фалкерсона* решает задачу нахождения максимального потока в транспортной сети. Идея алгоритма заключается в следующем. Изначально величине потока присваивается значение 0:

$f(u,v) = 0$ для всех $u, v \in V$. Затем величина потока итеративно увеличивается посредством поиска увеличивающего пути (путь от источника *s* к стоку *t*, вдоль которого можно послать больший поток). Процесс повторяется, пока можно найти увеличивающий путь.

Неформальное описание

1. Обнуляем все потоки. Остаточная сеть изначально совпадает с исходной сетью.

2. В остаточной сети находим любой путь из источника в сток. Если такого пути нет, останавливаемся.

3. Пускаем через найденный путь (он называется ***увеличивающим путём*** *или* ***увеличивающей цепью***) максимально возможный поток:

1. На найденном пути в остаточной сети ищем ребро с минимальной пропускной способностью $c_{\min}$.

2. Для каждого ребра на найденном пути увеличиваем поток на $c_{\min}$,

а в противоположном ему - уменьшаем на $c_{\min}$.

3. Модифицируем остаточную сеть. Для всех рёбер на найденном пути, а также для противоположных им рёбер, вычисляем новую пропускную способность. Если она стала ненулевой, добавляем ребро к остаточной сети, а если обнулилась, стираем его.

4. Возвращаемся на шаг 2.

Важно, что алгоритм не конкретизирует, какой именно путь мы ищем на шаге 2 или как мы это делаем. По этой причине алгоритм гарантированно сходится только для целых пропускных способностей, но даже для них при больших значениях пропускных способностей он может работать очень долго. Если пропускные способности вещественны, алгоритм может работать бесконечно долго, не сходясь к оптимальному решению (см. пример ниже).

Если искать не любой путь, а кратчайший, получится алгоритм Эдмондса-Карпа или алгоритм Диница. Эти алгоритмы сходятся для любых вещественных весов за время $O(|V||E|^2)$ и $O(|V|^2|E|)$ соответственно.

Формальное описание

Дан граф $G(V,E)$ с пропускной способностью $c(u,v)$ и потоком $f(u,v)$ для ребер из u в v. Необходимо найти максимальный поток из источника s в сток t. На каждом шаге алгоритма действуют те же условия, что и для всех потоков:

- $f(u,v) \leqslant c(u,v)$. Поток из u в v не превосходит пропускной способности.
- $f(u,v) = -f(v,u)$.
- для всех узлов u, кроме s и t. Поток не изменяется при прохождении через узел.

Остаточная сеть $G_f(V,E_f)$ — сеть с пропускной способностью $c_f(u,v)$ и без потока.

Вход Граф с пропускной способностью , источник и сток

Выход Максимальный поток из в

1. $f(u,v) \leftarrow 0$ для всех ребер (u,v)
2. Пока есть путь p из s в t в G_f, такой что $c_f(u,v) > 0$ для всех ребер $(u,v) \in p$:
   1. Найти $c_f(p) = \min\{c_f(u,v) \mid (u,v) \in p\}$
   2. Для каждого ребра $(u,v) \in p$
      1. $f(u,v) \leftarrow f(u,v) + c_f(p)$
      2. $f(v,u) \leftarrow f(v,u) - c_f(p)$

Путь может быть найден, например, поиском в ширину (алгоритм Эдмондса-Карпа) или поиском в глубину в $G_f(V,E_f)$.

Муравьиный алгоритм

Муравьи относятся к социальным насекомым, образующим коллективы. Коллективная система способна решать сложные динамические задачи по выполнению совместной работы, которая не могла бы выполняться каждым элементом системы в отдельности в разнообразных средах без внешнего управления, контроля или координации. В таких случаях говорят о роевом интеллекте, как о замысловатых способах кооперативного поведения, то есть стратегии выживания.

Одним из подтверждений оптимальности поведения муравьиных колоний является тот факт, что сеть гнёзд суперколоний близка к минимальному остовному дереву графа их муравейников.

Основу поведения муравьиной колонии составляет самоорганизация, обеспечивающая достижения общих целей колонии на основе низкоуровневого взаимодействия. Колония не имеет централизованного управления, и её особенностями являются обмен локальной информацией только между отдельными особями (прямой обмен – пища, визуальные и химические контакты) и наличие непрямого обмена, который и используется муравьиных алгоритмах. Таким образом, в общем случае рассматриваются слепые муравьи, не способные чувствовать близость пищи.

Непрямой обмен – стигмержи (stigmergy), представляет собой разнесённое во времени взаимодействие, при котором одна особь изменяет

некоторую область окружающей среды, а другие используют эту информацию позже, когда в неё попадают. Биологи установили, что такое отложенное взаимодействие происходит через специальное химическое вещество – феромон (pheromone), секрет специальных желёз, откладываемый при перемещении муравья. Концентрация феромона на пути определяет предпочтительность движения по нему.

Адаптивность поведения реализуется испарением феромона, который в природе воспринимается муравьями в течение нескольких суток. Мы можем провести некоторую аналогию между распределением феромона в окружающем колонию пространстве, и «глобальной» памятью муравейника, носящей динамический характер.

ЗАКЛЮЧЕНИЕ

Муравьиные алгоритмы могут быть успешно применены для решения сложных задач оптимизации. Основная идея, лежащая в основе алгоритмов муравьиной колонии, заключается в использовании механизма положительной обратной связи, который помогает найти наилучшее приближенное решение в сложных задачах оптимизации. То есть, если в данном узле муравей должен выбрать между различными вариантами и если фактически выбранные результаты будут хорошими, то в будущем такой выбор будет более желателен, чем предыдущий. Этот подход является многообещающим из-за его общности и эффективности в обнаружении очень хороших решений сложных проблем.

В данной бакалаврской работе приведены теоретические основы муравьиных алгоритмов оптимизации, проанализированы применения муравьиных алгоритмов, описаны решения муравьиными алгоритмами задачи коммивояжера и задачи оптимального распределения файлов в компьютерной сети. Решения задач были реализованы в среде программирования Delphi. Приведен сравнительный анализ результатов решения задачи коммивояжера с помощью муравьиного и генетических алгоритмов.