положения робота. При этом интервал «позади» робота, в расчет не принимается.

2. «Перенос» положения робота в вычисленную позицию центра перекрестка с отображением всех интервалов относительно новой системы координат. На рис. 3 такие интервалы изображены пунктирными линиями.
3. Кодирование бинарного образа перекрестка для НСВР с помощью преобразованных интервалов по описанному в разделе 2 правилу.
4. Сохранение или восстановление сохраненных координат перекрестка с помощью нейронной сети встречного распространения
5. Если на НСВР производился опрос сохраненных координат перекрестка, то произвести соответствующие изменения координат текущего положения робота.

4. Экспериментальные результаты

Тестирование системы локализации производилось на базе программной модели навигационной системы для управления автономным мобильным роботом. Параметры НСВР использовались следующие: количество перекрестков n =100, параметр корреляции p = 115 (96%) при длине бинарного входного образа s =120. Поскольку количество обрабатываемой сенсорной информации для системы локализации сокращено до необходимого минимального уровня, то процесс организации и модификации топологической карты не требует больших вычислительных затрат. В процессе тестирования полученной системы локализации, в условиях сложного лабиринта, был получен фактически стопроцентный результат. К полученным результатам, однако, необходимо сделать несколько комментариев:

1. В процессе моделирования системы рассматривались только идеальные сенсорные устройства – т.е. расстояние до препятствий «определялось» со 100% точностью.
2. Среда моделирования являлась одноагентной – т.е. в ней не присутствовали другие движущиеся объекты.
3. При тестировании системы локализации в реальной среде необходимо выбирать такой параметр корреляции p, чтобы система была толерантна к случайным помехам и подвижным препятствиям, и, в то же время, не происходило ложное определение координат перекрестка.

Заключение

В данной статье была рассмотрена система локализации положения робота в пространстве на базе нейронной сети встречного распространения. Выбор данного подхода был обусловлен эвристическим представлением о характере местности, которое позволяет из всего объема информации выделить для системы локализации только те отличительные признаки среды, по которым робот может быстро и эффективно определить свое местоположение. Реализация системы была построена на технологии топологических карт Кохонена, которые успешно себя зарекомендовали в задачах локализации и планирования маршрута движения для автономного транспортного средства.

Благодарности

Автор благодарит рецензентов за полезные замечания, оказанные в процессе обсуждения и написания данной статьи.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Heath-Pastore T., Everett H.R. and Bonner K. Mobile Robots for Outdoor Security Applications// American Nuclear Society 8th International Topical Meeting on Robotics and Remote Systems (ANS'99), 25-29 April, 1999 – Pittsburgh, PA, 1999.
2. Sukkarieh S., Nebot E.M. and Durrant-Whyte H.F. Achieving integrity in an INS/GPS navigation loop for autonomous land vehicle applications// IEEE International Conference on Robotics and Automation, 16-20 May 1998, Leuven, Belgium – 1998.
3. Thrun S. Particle Filters in Robotics// In Proceedings of Uncertainty in AI (UAI) – 2002.
4. Chong K.S. and Kleeman L. Mobile robot map building for an advanced sonar array and accurate odometry// International Journal Robotics Research – Jan 1999 – Vol 18, No. 1 – pp.20-36.
5. Chong K.S. and Kleeman L. Accurate odometry and error modelling for a mobile robot// IEEE International Conference on Robotics and Automation, Albuquerque, USA – April 1997 – pp.2783-2788.
6. Laird R.T, Everett H.R., Gilbreath G.A., Inderieden R.S. and Grant K.J. Early User Appraisal of the MDARS Interior Robot// American Nuclear Society 8th International Topical Meeting on Robotics and Remote Systems (ANS'99), 25-29 April, 1999 – Pittsburgh, PA, 1999.
7. Dellaert K.J., Fox D., Burgard W. and Thrun S. Monte Carlo Localization for Mobile Robots// IEEE International Conference on Robotics and Automation (ICRA99) – 1999.
8. Williams S.B., Dissanayake G. and Durrant-Whyte H.F. An Efficient Approach to the Simultaneous Localisation and Mapping Problem// International Conference on Robotics and Automation, May 2002 – Washington, DC. – vol.1 – pp.406-411.
9. Kohonen T. Statistical Pattern Recognition Revisited// Advanced Neural Computers/Eckmiller R. (Editor), Elsevier Science Publishers B.V. (North-Holland) – 1990.
10. Fritzke B. Growing Cell Structures – A Self-organizing Network for Unsupervised and supervised Learning// International Computer Science Institute/ Technical Report 93-026 – Berkeley, California.
11. Zimmer U.R. Self-Localization in Dynamic Environments // IEEE/SOFT International Workshop BIES '95, May 30-31, 1995 – Tokyo, Japan, 1995.

УДК 681.3.005.23

Прожерин И.Г.

ПРИБЛИЖЕННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА, ИСПОЛЬЗУЯ РЕШЕНИЕ ЗАДАЧИ О НАЗНАЧЕНИИ

1. ВВЕДЕНИЕ

Задача коммивояжера занимает центральное место среди труднорешаемых задач комбинаторной (дискретной) оптимизации. Все существенные идеи решения таких задач или были первоначально предложены для решения задачи коммивояжера, или, как правило, прошли проверку на этой задаче [1].

Задача коммивояжера является значимой в области дискретной оптимизации. Пока приемлемое точное решение по критерию времени и памяти возможно только для числа пунктов около 20. После многих неудачных попыток поиска глобального оптимума для большого числа точек становиться понятным, что точное решение связано со структурой рас-

Прожерин Игорь Геннадиевич, *ассистент каф. «Информатики и прикладной математики» Брестского государственного технического университета.*

Беларусь, БГТУ, 224017, г. Брест, ул. Московская, 267.

положения точек на плоскости. Предлагаемый алгоритм учитывает группировку точек на плоскости.

Задача коммивояжера имеет множество вариантов и в основном классифицируется типом матрицы расстояний (симметричная, несимметричная, метрическая, неметрическая) и по методу решения (точные и приближённые алгоритмы). К точным методам относятся метод полного перебора, динамического программирования, ветвей и границ, множителей Лагранжа, методы отсекающих плоскостей, композитные алгоритмы [2, 3]. К приближённым (эвристическим) методам относятся: асимптотические и генетические алгоритмы [3, 4, 5].

Будем рассматривать дискретную, нецелочисленную, симметричную или несимметричную задачу коммивояжера.

В развитии вычислительных алгоритмов решения трудных задач комбинаторной оптимизации и, конечно, в первую очередь задачи коммивояжера в последние годы четко обозначились две тенденции. Первая состоит в разработке алгоритмов, содержащих большое число вычислительных процедур, реализующих вычисления различных нижних оценок, набор высокоэффективных эвристик, использование множителей Лагранжа и отсекающих плоскостей, различные правила разбиения и ветвления, процедуры анализа и упрощения информации. Программные комплексы, реализующие такие алгоритмы, часто содержат программы управления вычислительным процессом и программы решения задач с участием человека в диалоговом режиме. Такие алгоритмы ориентированы на скрупулезный учёт специфики исходных данных задачи и призваны доставлять решение за минимальное время вычислительной системы. Конечно, в этом случае достаточно большими становятся расходы на алгоритмическое и программное обеспечение алгоритма [2,6].

Другая тенденция состоит в снижении расходов на программную реализацию алгоритмов. Исходная задача формулируется таким образом, что для ее решения используются только коммерческие пакеты или пакеты программ стандартного математического обеспечения ЭВМ [2].

Эвристические алгоритмы решения задачи коммивояжера являются основным инструментом решения практических задач. Появление эвристик обусловлено, в первую очередь, излишней чувствительностью точных алгоритмов по отношению к специфике задачи и наличию дополнительных условий. Эвристики представляют собой попытку учесть специфику задачи простыми средствами, создать прием, эффективный для решения задач с определённой особенностью [2,7,8,9].

2. ПОСТАНОВКА ЗАДАЧИ КОММИВОЯЖЕРА.

Предположим, что имеется ориентированный граф, каждое ребро которого имеет какой-то вес. Необходимо найти в нем гамильтонов цикл (то есть цикл, проходящий по всем вершинам ровно по одному разу) с минимальной суммой весов входящих в него ребер. Если провести аналогию с весом ребра и платой за проезд из одного города в другой, то получится, что коммивояжер хочет объехать все города какой-то области, затратив при этом минимальное количество денег на дорогу.

В матричной постановке задача звучит так, коммивояжер периодически посещает n городов. Расстояние между городами определяется числами c_{ij} из города i в j. Если прямого пути из города i в j нет, то c_{ij} принимает значение ∞. Из этих расстояний составляется матрица расстояний C диагональ, которой заполняется ∞.

Требуется составить маршрут передвижения коммивояжера, в котором:

- посещаются все города;
- ни один город, кроме начального, не посещается дважды;
- этот маршрут должен обладать минимальной суммарной протяжённостью.

Задача о коммивояжере сводиться к поиску гамильтонова контура минимальной длины [10].

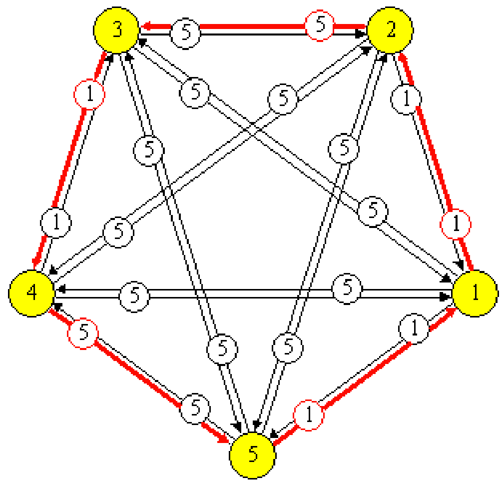

Рис. 1. Граф с решением задачи коммивояжера.

Например, для приведенного графа на рис.1, минимальным гамильтоновым циклом является выделенный цикл.

На данный момент считается, что эта задача не решается за полиномиальное время, т.е. эта задача принадлежит к классу так называемых NP- полных задач, точно так же как и задача о существовании в графе гамильтонова цикла. Однако это еще не доказано.

3. ПОСТАНОВКА ЗАДАЧИ О НАЗНАЧЕНИИ

Пусть задан двудольный граф G с двумя долями A и B, каждое ребро которого имеет какой-то вес. Для простоты будем считать, что в обеих долях одинаковое число вершин. Тогда задача состоит в том, чтобы выбрать в нем максимальное паросочетание с минимальным суммарным весом входящих в него ребер. Иными словами, мы должны каждой вершине доли A сопоставить вершину доли B, так чтобы, двум разным вершинам доли A соответствовали разные вершины доли B, и при этом сумма ребер, соединяющих эти пары вершин, была минимальна. Сам термин "задача о назначениях" возник из-за такой постановки этой задачи. Если вместо двудольного графа представить его матрицу смежности (а для двудольного графа элемент A[i, j] матрицы смежности это вес ребра, соединяющего вершину номер i из первой компоненты и вершину j из второй компоненты), то мы должны будем выбрать из каждой строки и каждого столбца ровно по одному элементу, так, чтобы сумма выбранных элементов была минимальна. Если считать вершины первой доли рабочими, вершины второй доли - работами, а ребра - ежемесячной платой рабочему за соответствующую работу, то задача сводится к тому, чтобы назначить каждого рабочего на какую-либо работу так, чтобы платить им всем как можно меньше. Метод решения задачи о назначениях называется "Венгерским методом". [11]

4. ТОЧНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА НА ОСНОВЕ РЕШЕНИЯ ЗАДАЧИ О НАЗНАЧЕНИИ

Рассмотрим граф, для которого необходимо решить задачу коммивояжера. Рассмотрим его матрицу смежности и попробуем решить для нее задачу о назначениях. Для примера возьмем все тот же граф на рис.1. Для его матрицы смежности задача о назначениях выделит следующие элементы (рис.2)

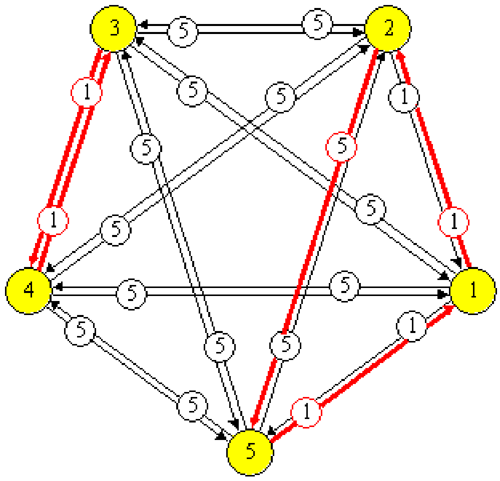

Рис. 2. Граф с отмеченным решением задачи о назначении.

Теперь, если посмотреть внимательно, то видно следующее: если сопоставить выделенным элементам ребра графа, то получится, что из каждой вершины графа выходит ровно одно ребро и выходит ровно одно ребро. То есть граф разобьется на непересекающиеся циклы.

И так, сделаем вывод - задача о назначениях решает задачу несколько схожую с задачей коммивояжера. Если задача коммивояжера разбивает граф на один цикл так, чтобы суммарный вес ребер этого цикла был минимален, то задача о назначениях разбивает граф на несколько циклов с выполнением того же условия.

Теперь построим следующий алгоритм. Сначала решим задачу о назначениях для матрицы смежности графа. Если она разобьет граф на один цикл, то алгоритм закончен, решение найдено. Если нет, то сделаем следующую операцию: по отдельности выкинем из графа ребра одного из циклов. Тогда получится несколько новых графов, каждый из которых обладает тем свойством, что ни для одного из них задача о назначениях не даст больше такого же решения. Кроме того гамильтонов цикл для исходного графа не мог содержать все ребра удаленного цикла, а следовательно он является также и гамильтоновым путем для одного из получившихся графов. Следующий шаг понятен, нужно применить этот же алгоритм для каждого из новых графов и выбрать наилучший из получившихся ответов. Очевидно, этот алгоритм никогда не зациклится, т.к. на каждом шаге количество ребер в графах уменьшается.

Реализация этого алгоритма имеет некоторые особенности. Поскольку нам нужно визуализировать процесс рекурсии, прибегнем к помощи такого объекта как стек. В него мы будем класть получающиеся новые графы и соответственно будем брать их из него, когда это потребуется. Откидывая некоторые лишние варианты, мы можем ускорить работу алгоритма. Если, к примеру, с помощью задачи о назначениях для графа найдено решение, превышающее по длине уже найденный коммивояжеров путь, то рассматривать такой граф нет больше смысла. Мы также можем сразу откидывать графы, которые вообще нельзя разбить на циклы.

Общая схема решения задачи коммивояжера точным алгоритмом, используя решение задачи о назначении:
1) возьмем исходный граф или граф из стека. Если стек пуст, переходим к шагу 6;
2) первоначальная проверка на существование разбиения графа на циклы. Проверим, есть ли вершины, из которых не выходит ни одно ребро или не входит ни одного ребра. Если есть, то этот граф больше не рассматриваем, т.е. переходим к шагу 1;
3) решим задачу о назначениях для матрицы графа;
4) если решение задачи о назначении является решением задачи коммивояжера, то запоминаем найденный путь и переходим к шагу 1. Если нет, то в случае, когда суммарная длина найденных циклов превышает длину уже найденного пути, переходим к шагу 1, в противном же случае переходим к шагу 5.
5) по отдельности выкидываем из графа ребра одного из циклов, кладем получившиеся графы в стек, переходим к шагу 1;
6) конец аглоритма. Выводим длину минимального найденного пути, или сообщаем, что такой не найден.

5. ПРИБЛИЖЕННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА НА ОСНОВЕ РЕШЕНИЯ ЗАДАЧИ О НАЗНАЧЕНИИ

Точный алгоритм решения задачи коммивояжера, используя решение задачи о назначении, имеет трудоемкость равную приблизительно $o(\sim 10n^2)$. Метод ветвей и границ имеет трудоемкость того же порядка. Для решения задачи коммивояжера данным алгоритмом требуется объем памяти приблизительно $\sim (2n^3)$ для получения решения.

В некоторых прикладных задачах, когда $n \geq 1000$, такие параметры не приемлемы. Для улучшения параметров трудоемкости и требуемой памяти для решения, можно предложить приближенный алгоритм решения задачи коммивояжера, используя решение задачи о назначении.

Приближенный алгоритм, отличается от точного лишь тем, что не нужно делать перебор по всем ребрам каждого цикла, необходимо выбрать в каждом цикле одно ребро с максимальным весом и удалить его, а затем пересчитать задачу о назначении.

Общая схема решения задачи коммивояжера приближенным алгоритмом, используя решение задачи о назначении:
1) решается задача о назначении, в результате чего в графе выделяются замкнутые циклы;
2) если кол-во циклов равно 1 то решение задачи окончено, иначе переходим к шагу 3);
3) в каждом полученном замкнутом цикле необходимо найти ребро с максимальным весом и удалить его;
4) переходим к шагу 1).

6. СРАВНЕНИЕ РАЗЛИЧНЫХ МЕТОДОВ

Определение параметров эффективности алгоритма – одна из важнейших задач. Как определить свойства алгоритма? Для начала необходимо выбрать критерии оценки алгоритмов. Основными критериями являются трудоемкость, использование памяти, качество решения. Поэтому сравнение алгоритмов будем рассматривать в разрезе этих критериев.

Существует много различных методов получения оценок трудоемкости алгоритмов. Один из таких методов представляет собой следующую процедуру:
1) ранжируются все операции в алгоритме;
2) определяются группы операций по трудоемкости;
3) каждой группе команд приписывается определенный вес;
4) определяется количество выполнений каждой группы команд;
5) значения количества выполнения операций каждой группы приводятся к одной размерности и порядку.

Определение объема занимаемой памяти – задача тривиальная. Для этого необходимо определить размер памяти, занимаемый одним элементом и определить необходимое количество элементов. Умножая эти значения, получим объем требуемой памяти для решения задачи [11-13].

Таблица 1. Теоретическое сравнение затрачиваемых ресурсов

Наименование алгоритма	**Трудоёмкость**	**Память**	**Отклонение от оптимального значения, %**
Ветвей и границ	$o(\sim 7n^2)$	$\sim(2n^3)$	0
Метод решения задачи коммивояжера, используя решение задачи о назначении (точный)	$o(\sim 10n^2)$	$\sim(2n^3)$	0
Метод решения задачи коммивояжера, используя решение задачи о назначении (приближенный)	$o(\sim n^2)$	$\sim(2n^2)$	10

Теоретическое сравнение рассмотренных алгоритмов по вычислительной трудоёмкости и количеству памяти представлено в табл.1. Результаты получения оценок по методу ветвей и границ и точному методу решения задачи коммивояжера, используя решение задачи о назначении, приведены в [1-4,11-13].

Анализируя данные в табл.1. видно, что алгоритмы предложенные ранее (ветвей и границ и точный алгоритм решения задачи коммивояжера, используя решение задачи о назначении), либо требуют использования большого объёма памяти порядка $\sim(2n^3)$, либо имеют трудоёмкость большую в 7-10 раз.

При оценке алгоритмов по взаимоисключающим критериям не трудно понять, что проигрывая в одном можно выиграть в другом и наоборот. Идеального варианта быть не может. Качество приближенного алгоритма можно получить, решая различные задачи и оценивая их решение относительно точных методов. Отклонение, указанное в табл.1 для приближенного алгоритма решения задачи коммивояжера, используя решение задачи о назначении, было получено экспериментальным путем. Случайным образом генерировалась задача коммивояжера и решалась различными методами, при этому определялась длина пути и отклонение в процентах для приближенного метода относительно точных. Для приближенного алгоритма решения задачи коммивояжера, используя решение задачи о назначении, отклонение от оптимального решения, как показал эксперимент, не превышает 10%.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Меламед И. И., Сергеев С. И., Сигал И. Х., Задача коммивояжера. Вопросы теории // Автоматика и телемеханика. – М.: Наука, 1989. №9. с.3-33.
2. Меламед И. И., Сергеев С. И., Сигал И. Х., Задача коммивояжера. Точные методы // Автоматика и телемеханика. – М.: Наука, 1989. №10. с.3-29.
3. Меламед И. И., Сергеев С. И., Сигал И. Х., Задача коммивояжера. Приближенные алгоритмы // Автоматика и телемеханика. – М.: Наука, 1989. №11. с.3-26.
4. Few L., The Shortest Path and the Shortest Road Through n Points, Mathematica, 2, 1955, 141]
5. Новиков Ф. А., Дискретная математика для программистов, СПб.: Питер, 2000.–304 с.:ил.
6. Перепелица В. А., Гимади Э. Х., К задаче нахождения минимального гамильтонова контура на графе со взвешенными дугами, Сб. «Дискретный анализ», Новосибирск, вып. 15, 1969, 57-65.
7. Kaluga V.V., Muravjev S. A., Siridonov S. V., Telyatnikov R. V., Application of genetic algorithms for solutions of the task is frequent – territorial plannings group radio electronic equipment, International Conference of Neural Networks and Artificial Intelligence ICNNAI'99|Proceedings. Editted by Vladimir Golovko, - Brest: BPI, 1999, 224p.
8. Гимади Э. Х., Перепелица В. А., Асимптотический подход к решению задачи коммивояжера, Сб. «Управляемые системы», Новосибирск, вып. 12, 1974, 35-45.
9. Гимади Э. Х., Перепелица В. А., Статистически эффективный алгоритм выделения гамильтонова контура (цикла), Сб. «Дискретный анализ», Новосибирск, вып. 22, 1973, 15-28.
10. Новиков Ф. А., Дискретная математика для программистов, СПб.: Питер, 2000.–304 с.
11. Кристофидес Н., "Теория графов. Алгоритмический подход", М.: Мир, 1978
12. Гэри М., Джонсон Д., "Вычислительные машины и труднорешаемые задачи" :пер. с англ. – М.:Мир, 1982. – 416 с.
13. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985.

УДК 681.326.7.74

Шуть В.Н., Муравьёв Г.Л., Чуль Д.Д.

ОПТИМИЗАЦИЯ ЧИСЛА КАНАЛОВ СИГНАТУРНОГО АНАЛИЗАТОРА

ВВЕДЕНИЕ

Одним из основных методов тестирования цифровых узлов (ЦУ) в настоящее время является сигнатурный анализ, предложенный фирмой Hewlett – Packard [1].

При этом анализатор HP5004A разработанный фирмой, а также аналогичные приборы других изготовителей [2,3] рассматриваются, как средства сервисного контроля и ремонта ЦУ у потребителя или в центрах обслуживания и ремонта цифровой техники.

Попытка применить подобные приборы в серийном производстве, т.е. при массовом контроле многовыходных ЦУ не дает желаемого эффекта. Действительно, контроль n - выходного ЦУ одноканальным сигнатурным анализатором (СА) приводит к увеличению времени контроля в n раз.

В [4-10], [14] предложены методы и схемы синтеза многоканальных СА, позволяющих за один цикл тестовой инициализации ЦУ осуществить параллельную свертку в сигнатуру всех n выходов ЦУ. Однако синтез подобных СА для большого числа выходов ЦУ (n) трудоемок и требует больших аппаратных затрат.

К тому же каскадирование большого числа элементов свертки по mod2 согласно методике [8] снижает быстродей-

Шуть Василий Николаевич*, к.т.н., доцент каф. ИИТ Брестского государственного технического университета.*

Муравьёв Геннадий Леонидович, *к.т.н., профессор ИСЗ.*

Чуль Дмитрий Дмитриевич*, студент V курса, специальность ЭВМ Брестского государственного технического университета.*

Беларусь, БГТУ, 224017, г. Брест, ул. Московская, 267.