

СЕКЦИЯ 1

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Среда, 28 апреля 2010 г., читальный зал преподавателей (ауд.232), гл. корпус МГТУ им.Н.Э.Баумана.

Начало в 10.00.

Председатель: профессор, д.т.н. Норенков И.П.

Руководитель экспертной комиссии: к.т.н., доцент Федорук В.Г.

Ученый секретарь: к.т.н., доцент Власов А.И.

Экспертная комиссия:

БОЖКО А.Н., ВОЛОСАТОВА Т. М., ЖУК Д. М., МАНИЧЕВ В. Б., МАРТЫНЮК В. А., ТРУДОНОШИН В. А., ГРОШЕВ С. В., КНЯЗЕВА С. РОДИОНОВ С. В., ФЕДОРУК Е., ИВАНОВ В.В.

В рамках работы секции будет осуществлен первый этап конкурсного отбора проектов в проектный инкубатор МГТУ им.Н.Э.Баумана. Критериями конкурсного отбора участников молодежного научно-инновационного конкурса являются:

1. Уровень инновационности идеи (предложения, метода, способа ...).
 - 1.1. Идея должна быть новой, впервые сформулированной именно самим номинантом.
 - 1.2. Уровень наукоемкости тем выше, чем более:
 - основательны научные исследования, в результате которых она появилась;
 - основательны дальнейшие научные исследования, необходимые для ее реализации.
 - 1.3. Техническая значимость тем выше, чем большее влияние ее реализация окажет на уровень техники. «Пионерные» идеи (изобретения) – пенициллин, лазер, синтез алмазов - открывают новые отрасли науки и техники. Идеальная по инновационности идея неожиданна для рынка. Поэтому она им не может быть сейчас востребована, она сама формирует новую потребность и нишу рынка.
Высокий технический уровень имеют решения, например, многоотраслевого использования.
Оригинальные технические решения дают новые принципы решения известной задачи.
Есть решения, позволяющие решить проблему еще одним, дополнительно к известным, способом. Но и в этом случае, если уже известно 10 способов решения задачи, то новый способ может и не давать существенных преимуществ в решении задачи или давать их только в очень ограниченном по масштабам применения числе случаев.
А может быть и просто рацпредложение.
 - 1.4. Масштабность использования предложения тоже может сильно различаться – от решения локальной задачи одного местного потребителя до ...
Грубый способ оценки масштабности экспертами – возможный полный объем производства продукта на основе данного предложения – 106, 107, 108, 109, 1010, 1011 рублей
 - 1.5. Срок превращения идеи в конечный продукт с выходом его на рынок: новизна, рискованность идеи, объем необходимых научных исследований не позволяют уложиться в 2-3 года, но и не требуют 10-15 лет
 - 1.6. Идея тем актуальней, чем меньше вероятность того, что за 5-7 лет в результате научных исследований появятся и «раскрутятся» до продукта другие более эффективные пути решения задачи.

Победитель сам организует работу по привлечению необходимого финансирования. Программа оказывает ему поддержку на начальном этапе работы над его идеей.

ИЗГОТОВЛЕНИЕ ШАБЛОНА С НАПРАВЛЯЮЩИМИ ДЛЯ СВЁРЛ С ЦЕЛЮ УПРОСТИТЬ ПРОЦЕСС СВЕРЛЕНИЯ ОТВЕРСТИЙ В ЧЕЛЮСТИ ПАЦИЕНТА ВО ВРЕМЯ УСТАНОВКИ ЗУБНЫХ ИМПЛАНТАТОВ

А. Михайловс

Научный руководитель к.т.н, доцент Д.М. Жук
МГТУ имени Н.Э.Баумана, кафедра РК6, Москва, Россия

A TEMPLATE WITH GUIDING HOLES FOR DRILL PRODUCTION TO SIMPLIFY HOLES DRILLING PROCESS IN A PATIENT JAW AT A TIME OF PLACING A TOOTH IMPLANT

A. Mihailovs

Supervisor D. Zhuk

Bauman Moscow State Technical University

Аннотация

Задачей является изготовление пластикового шаблона с нижней стороны частично повторяющего поверхность челюсти пациента, который бы помог просверлить отверстия в челюсти в нужном месте под нужным углом. Без шаблона такие отверстия может сделать только очень опытный хирург, тогда как шаблон допускает более широкий круг специалистов. Задача является экономически выгодной, так как найти опытного хирурга бывает непросто.

Abstract

The goal is to produce a plastic template which could help drilling a hole in a jaw at right place with right angle. Without a template holes like this could make only very skillful surgeons but a template allows more wide range of professionals. The task is economically profitable because it could be difficult to find a skillful surgeon.

Для реализации данной задачи было выявлено несколько подходов:

- использование готовых программных продуктов и немного программирования для доводки промежуточных результатов
- создание собственного программного комплекса для выполнения всех промежуточных стадий

С целью ускорить получение результата был выбран более простой подход к решению задачи – использовались уже готовые программные продукты, а результат выполнения некоторых подогнать для входных данных последующих.

Для начала необходимо сделать снимок челюсти человека с помощью томографа. Воспользуемся малодозным, так как он более распространен. Имплантолог крепит на челюсть человека контрастный материал, который задает направление и место сверления и делает снимок. Получаем некоторое количество файлов формата DICOM, из которых мы и будем выделять нужную нам поверхность.

DICOM (Digital Imaging and Communications in Medicine, цифровые изображения и обмен ими в медицине) - это индустриальный стандарт для передачи радиологических изображений и другой медицинской информации между компьютерами, опирающийся на стандарт Open System Interconnection (OSI), разработанный Международной организацией по стандартам (International Standards Organization, ISO). Стандарт DICOM описывает "паспортные" данные пациента, условия проведения исследования, положение пациента в момент получения изображения и т.п.

Существует программный продукт, называемый Amira, который способен читать данные DICOM и преобразовывать построенную на основе полутоновых изображений воксельную модель в единственный формат – VRML.

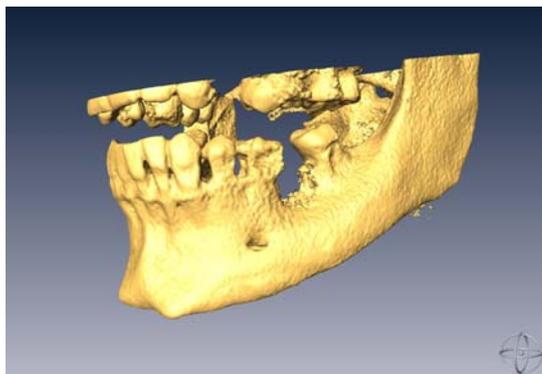


Рис. 1. Воксельная модель челюсти

Воксел (в разговорной речи воксель, [англ. Voxel](#) — образовано из слов: объёмный ([англ. volumetric](#)) и [пиксел \(англ. pixel\)](#)) — элемент объёмного изображения, содержащий значение элемента [растра в трёхмерном пространстве](#). Воксели являются аналогами пикселей для трёхмерного пространства. Воксельные модели часто используются для визуализации и анализа [медицинской](#) и научной информации.

Программный продукт Amira платный, но написан специально для работы с форматом DICOM, поэтому никаких нареканий не вызвал. Процесс работы достаточно прост. Для начала загрузим данные томографа – для этого нажимаем кнопку “Open Data...”. В открывшемся диалоговом окне выбираем все снимки томографа (файлов может быть много) и жмем загрузить. Теперь имеем набор снимков, по которым нужно построить 3д модель. Построение 3д модели происходит после нажатия на кнопку “Isosurface” и выставления необходимого Threshold (отклонение от максимальной контрастности вокселей модели). От Threshold зависит количество точек и, соответственно, качество модели. Задача определения оптимального Threshold решается отдельно для каждого набора снимков. Теперь осталось только экспортировать данные в формат VRML. Для этого нажмем кнопку “VRML – Export” и выберем место сохранения. В итоге имеем файл с описанием всех точек и связей между ними.

В качестве редактора 3д сеток выбираем программу 3ds max, так как это один из наиболее простых в освоении и мощных редакторов на данный момент. 3ds max позволяет не только редактировать поверхности, но и импортировать различные форматы данных, что нам и необходимо для работы.

Импортировать формат VRML программный продукт 3ds max позволяет, начиная с версии 2010, но, для моделей с огромным количеством вершин этот процесс чрезвычайно длителен, и, зачастую, заканчивается неудачей. Поэтому мы вынуждены использовать дополнительное конвертирование в формат 3DS, хорошо понимаемый программой 3ds max. Конвертирование проходит в несколько этапов:

- загрузка VRML модели в память
- разделение модели на слои по максимальному количеству вершин в каждом (связано с особенностями 3DS формата)
- сохранение модели в формат 3DS

Теперь, воспользовавшись сторонним конвертером, в нашем распоряжении имеется 3DS файл с информацией о модели челюсти, который пока что не пригоден к использованию из-за возникших трудностей и проблем на этапе создания. Все эти проблемы можно исправить в программе 3ds max с помощью MaxScript, написав соответствующую программу.

MaxScript – это язык программирования 3ds max, позволяющий автоматизировать монотонные процессы, на которые обычно уходит много времени. Если одни и те же операции постоянно повторяются, значит необходимо использовать MaxScript.

Для нормальной работы с 3д изображением необходимо исправить все проблемы, возникшие на этапе конвертации. Такими проблемами являются:

- несоответствие систем координат DICOM данных и 3d max сцены (модель получается повернутой на -90° по оси X)
- раздробленность модели на слои (из-за особенностей формата 3ds имеем модель, разбитую на слои, для объединения которых необходимо слить совпадающие в пространстве вершины)
- модель отображается без сглаживания (для улучшения зрительного восприятия модель можно сгладить, добавив соответствующий модификатор)

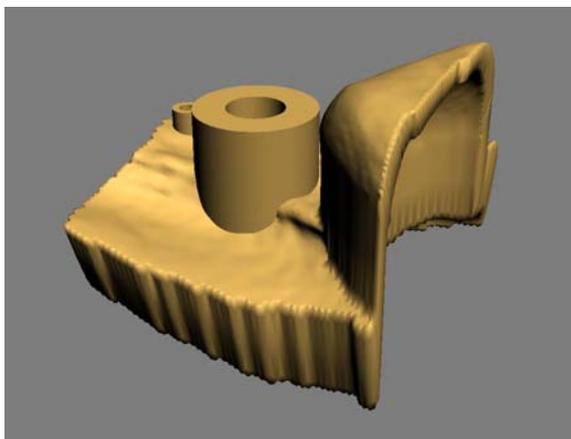


Рис. 2. Полученный шаблон

Теперь, в нашем распоряжении имеется цельная модель, готовая к изменениям. На этом этапе нужно создать пластиковый шаблон, который бы своей нижней поверхностью повторял верхнюю поверхность челюсти пациента. Сделать это человеку, работающему с программой 3ds max, не составит труда. Для воплощения 3д модели в жизнь нужно лишь распечатать её на 3д-принтере. Для этого конвертируем нашу модель в бинарный STL и отправляем на печать.

В результате проделанной работы мы получили пластиковый шаблон, который поможет имплантологу при сверлении отверстий в челюсти.

По идее, конечной целью данного проекта является написание такого программного комплекса, который бы выполнял все описанные выше промежуточные функции самостоятельно, без использования сторонних программных продуктов. Поэтому, найденное решение является временным и, может существовать до тех пор, пока такой комплекс не будет написан. Предположительно, комплекс должен будет включать в себя такие этапы:

- чтение данных DICOM (в этом поможет SDK)
- построение изоповерхности используя методы трехмерной реконструкции (то, чем сейчас занимается Amira)
- редактор трехмерных сеток на основе OpenGL или Direct3D
- конвертер полученной модели в формат STL для отправки в печать

Реализовав предложенную концепцию, ручная работа останется только на этапе проектирования самого шаблона, но и её можно минимизировать.

ПРОГРАММНЫЙ МОДУЛЬ СРАВНЕНИЯ ЭЛЕКТРИЧЕСКИХ СХЕМ

Ентин Д.Б.

Научный руководитель к.т.н., доцент Федорук В.Г.

МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Россия

THE PROGRAM MODULE OF COMPARING ELECTRICAL SCHEMES

Dmitry B. Entin

Supervisor Ph.D., Associate professor Vladimir G. Fedoruk

Moscow State Technical University n.a. N.E. Bauman

Аннотация

В данной статье рассмотрены структура модуля сравнения электрических схем, состоящих из двух- и четырёхполюсных элементов, и этапы решения задачи установления факта эквивалентности сравниваемых схем. В тексте приведены описания синтаксиса исходных данных модуля и последовательности выполняемых действий, а также подробно рассмотрены алгоритмы, применяемые на основных этапах решения задачи.

Abstract

The given article considers the structure of the module of comparing electric schemes consisting of bipolar and quadropolar elements and the stages of solving the problem of the establishment of the equivalence of the compared schemes. The text contains the description of the syntax of the initial data of the module and of the sequence of the operations performed, as well as the detailed examination of the algorithms used at the basic stages of the solution of the problem.

Данный модуль разрабатывается для решения проблемы верификации топологии микросхем, а так же для возможности расширенной проверки навыков у студентов по решению задач построения эквивалентных схем для различных систем (с помощью системы онлайн-тестирования).

Входная информация для модуля – текстовые описания сравниваемых схем, которые преобразуются в массивы элементов. Пример синтаксиса:

< Идентификатор элемента > < Список узлов > < Список значений >.

Идентификатор элемента состоит из обозначения типа элемента и его номера. Используемые обозначения типа элемента (заглавные буквы латинского алфавита): двухполюсники - E (источник типа E), C (конденсатор), R (резистор), L (индуктивность), I (источник типа I); четырёхполюсники – TF (трансформатор), GT (гиратор).

В зависимости от типа элемента зависит длина списка узлов (она составляет два и четыре узла для двухполюсников и четырёхполюсников соответственно) и длина списка значений (ни одного или одно значение – для двухполюсников, одно или два значения – для четырёхполюсников).

Узлы нумеруются цифрами, от нуля, без пропусков. Нулём всегда обозначается базовый узел («земля»). Для источников (элементов типа I и E), а так же для четырёхполюсников, важна последовательность указания узлов. Направление у источников – от узла, указанного первым, к узлу, указанному вторым. У трансформаторов выходы с левой стороны соответствуют источнику типа I, с правой стороны – источнику типа E.

Первый этап сравнения – преобразование четырёхполюсных элементов.

Получив массивы элементов, модуль преобразует четырёхполюсные элементы типа TF (трансформаторы) и GT (гираторы) в пары двухполюсных элементов. Новые элементы добавляются в существующие массивы. Трансформаторы преобразуются в источник типа I (который получает идентификатор вида I_x <номер трансформатора>) с узлами №1 и №2 и источник типа E (который получает идентификатор вида E_x <номер трансформатора>) с узлами №3 и №4. Значения параметров трансформатора копируются в значения параметров этих элементов. Гираторы преобразуются по той же схеме на два источника типа I, которые

получают аналогичные идентификаторы и пары узлов №1-№2 и №3-№4. Значения параметров так же копируются.

Второй этап сравнения – построение графов по схемам.

Имея подготовленный список элементов, модуль составляет графы по схемам. Графы хранятся в виде матриц инцидентности (строкам соответствуют элементы, столбцам – узлы схем). Элементы заносятся в матрицу в порядке приоритетов их типов. Была выбрана следующая последовательность: $E \rightarrow C \rightarrow R \rightarrow L \rightarrow I$. Так же строки матрицы инцидентности упорядочиваются по убывания значений параметров элементов (но без нарушения порядка приоритетов). Такой подход необходим для облегчения выделения остовных деревьев графов, а так же для гарантии попадания в них одинаковых элементов. Трансформаторы и гираторы на этом этапе игнорируются, так как информация о них содержится в соответствующих двухполюсниках.

Третий этап сравнения – выделение остовных деревьев.

Для выделения остовных деревьев модуль берёт необходимое для покрытия всех столбцов количество строк матрицы инцидентности, начиная с первой. Затем производится проверка деревьев на связность. В случае обнаружения узлов, не связанных с остальными, дерево достраивается необходимыми дугами.

Корнем дерева считается узел с номером 0 (базовый узел, «земля»).

Четвёртый этап сравнения – проверка остовных деревьев на изоморфизм.

На данном этапе начинается собственно сравнение схем. Сначала производятся базовые проверки на равенство количества узлов, двухполюсных и четырёхполюсных элементов сравниваемых схем, а так же количество элементов, вошедших в остовные деревья.

В случае положительного результата этих проверок, выполняется проверка изоморфности деревьев по алгоритму, описанному ниже. В ходе работы этого алгоритма вычисляется количество уровней дерева (групп узлов, расположенных на одинаковом расстоянии от корня), которое так же должно быть одинаковым у эталонной и пользовательской схем.

В случае невыполнения базовых условий, модуль принимает решение о неравенстве эталонной и пользовательской схем.

Применяется алгоритм, описанный в [1]. Он выполняет процедуру проверки за время, пропорциональное числу узлов в деревьях. Алгоритм приписывает целые числа узлам двух сравниваемых деревьев, начиная с узлов последнего уровня и двигаясь вверх до корней, так что деревья изоморфны тогда и только тогда, когда их корням приписано одно и то же число.

Схема алгоритма:

1. Приписать число 0 всем листьям деревьев T_1 и T_2 .

2. Индукция. Предположим, что целые числа уже приписаны всем узлам, находящимся в деревьях T_1 и T_2 на уровне $[i-1]$. Пусть L_1 - список узлов уровня $[i-1]$ в дереве T_1 , расположенных в порядке не убывания приписанных им чисел, а L_2 - соответствующий список для T_2 .

3. Приписать непомяченным узлам уровня $[i]$ в дереве T_1 кортеж целых чисел следующим образом: просмотреть список L_1 слева направо и для каждого узла v и из L_1 взять число, ему приписанное, в качестве очередной компоненты кортежа, который ставится в соответствии отцу узла v .

После выполнения этого шага каждому непомяченному ранее узлу w уровня $[i]$ в дереве T_1 будет поставлен в соответствие кортеж (i_1, i_2, \dots, i_k) где i_1, i_2, \dots, i_k – целые числа, приписанные сыновьям узла w и расположенные в порядке не убывания. Обозначим через S_1 последовательность кортежей, построенных для узлов уровня $[i]$ в дереве T_1 .

4. Повторить шаг 3 для T_2 , пусть S_2 - последовательность кортежей, построенных для узлов уровня $[i]$ в дереве T_2 .

5. Упорядочить S_1 и S_2 в лексикографическом порядке. Пусть соответственно S'_1 и S'_2 - упорядоченные последовательности кортежей.

6. Если S'_1 и S'_2 не совпадают, то остановиться; в этом случае исходные деревья не изоморфны. В противном случае приписать число 1 тем узлам уровня $[i]$ в дереве T_1 , которые представлены первым кортежем в S'_1 , число 2 - вторым отличающимся кортежем (имеется в виду не кортеж, стоящий на втором месте в списке S'_1 , а тот кортеж, который окажется на втором месте после вычёркивания из S'_1 всех повторений), и т.д.

Так как эти целые числа приписаны узлам уровня $[i]$ в дереве T_1 , построить список L_1 из узлов, которым таким способом приписаны числа. Добавить к началу списка L_1 все листья дерева T_1 , расположенные на уровне $[i]$. Пусть L_2 - соответствующий список узлов дерева T_2 .

Эти два списка теперь можно использовать для приписывания кортежей узлам уровня $[i+1]$ с помощью процедуры на шаге 3.

7. Если корням деревьев T_1 и T_2 приписано одно и то же число, то T_1 и T_2 изоморфны.

На рисунке 1 иллюстрируется пример приписывания чисел и кортежей узлам двух изоморфных деревьев.

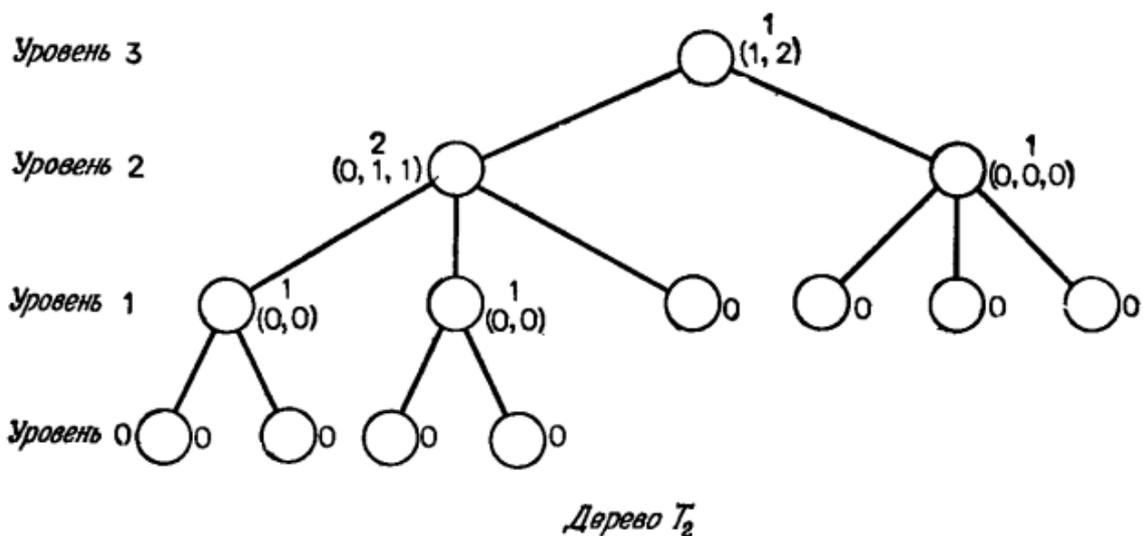
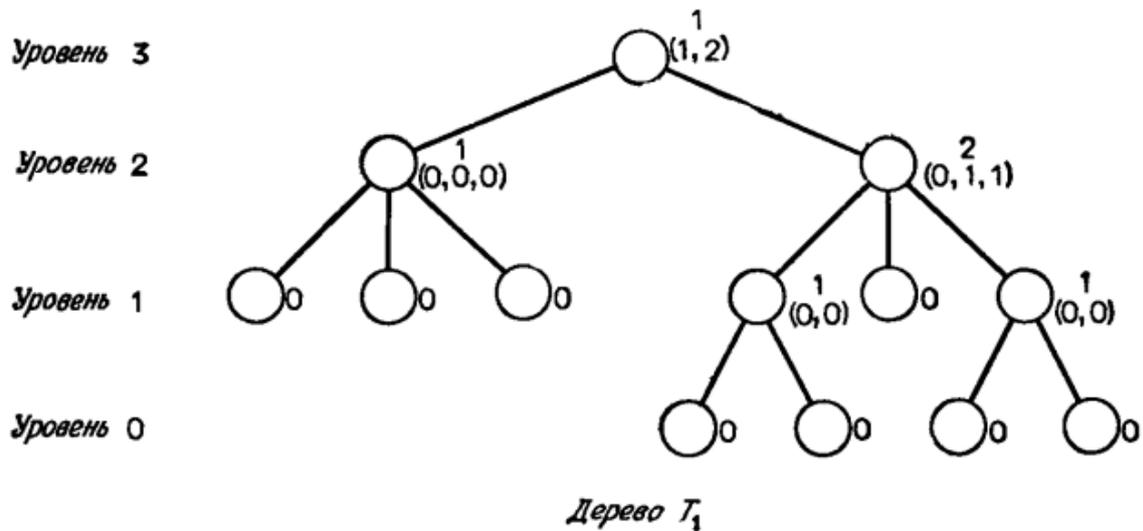


Рис. 1. Пример пометки вершин.

Пятый этап сравнения – поиск соответствия номеров узлов изоморфных деревьев.

В случае, если факт изоморфности остовных деревьев эталонной и пользовательской схем доказан, мы можем приступить к поиску соответствия нумерации узлов.

Модуль считает, что два узла эквивалентны, если выполняются все нижеперечисленные условия:

- Родительские узлы уже сопоставлены и эквивалентны.
- Узлы находятся на одном уровне.
- Узлы имеют равное количество дочерних узлов.
- Должно существовать соответствие между элементами эталонной и пользовательской схем, содержащими сравниваемые узлы.

При обнаружении неоднозначностей применяется модификация алгоритма поиска с возвратом, используемого в интерпретаторе языка Пролог (описан в [2]).

Описание базового алгоритма:

В том месте программы, где возможен выбор нескольких вариантов, интерпретатор языка Пролог сохраняет в специальный стек точку возврата для последующего возвращения в эту позицию. Точка возврата содержит информацию, необходимую для возобновления процедуры при откате. Выбирается один из возможных вариантов, после чего продолжается выполнение программы.

Во всех точках программы, где существуют альтернативы, в стек заносятся указатели. Если впоследствии оказывается, что выбранный вариант не приводит к успеху, то осуществляется откат. Выбирается последняя из имеющихся в стеке точек программы, где был выбран один из альтернативных вариантов. Выбирается очередной вариант, программа продолжает свою работу. Если все варианты в точке уже были использованы, то регистрируется неудачное завершение и осуществляется переход на предыдущую точку возврата, если такая есть. При откате все связанные переменные, которые были означены после этой точки, опять освобождаются.

На выходе этого этапа модуль имеет таблицу соответствия. Если данную таблицу построить не удалось, то модуль принимает решение о неравенстве эталонной и пользовательской схем.

Шестой этап сравнения – проверка элементов на соответствие и вынесение решения об эквивалентности схем.

На данном этапе узлы одной из схем переименовываются в соответствии с таблицей соответствия, и модуль проводит окончательную проверку схем на эквивалентность. Элементы эталонной схемы последовательно сравниваются с элементами пользовательской схемы, при обнаружении соответствия элементы выбывают из дальнейшего поиска.

Если соответствие найдено, модуль принимает решение об эквивалентности схем. В противном случае, модуль пытается перестроить таблицу соответствия узлов, используя информацию о неоднозначностях, полученную на предыдущем этапе. Если проверены все варианты таблицы соответствия узлов, то модуль принимает решение о неравенстве эталонной и пользовательской схем.

Литература

1. Ахо, А., Хопкрофт, Дж., Ульман, Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979. – 536 стр.
2. Стерлинг, Л., Шапиро, Э. Искусство программирования на языке Пролог. --- М.: Мир, 1990. --- 333 стр.

ОБЗОР МЕТОДОВ СИНТЕЗА ПОСЛЕОПЕРАЦИОННОГО ОБЛИКА ПАЦИЕНТА

Демина А.И.

Научный руководитель: Жук Дмитрий Михайлович, к.т.н., доцент
МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Россия

THE REVIEW OF METHODS OF SYNTHESIS POSTOPERATIVE OF SHAPE OF THE PATIENT

Demina Anastasia Ygorevna

Guk Dmitry Mihajlovich, PhD, the senior lecturer
Moscow State Technical University n.a. N.E. Bauman

Аннотация

В рамках данной работы рассматривается решение задачи синтеза послеоперационного облика пациента при травмах и деформациях челюстного отдела лицевого скелета.

В качестве предметной области, необходимой к изучению, выступают данные о строении лицевого скелета человека, существующие методики получения облика пациента, данные о цефалометрических точках, плоскостях и измерениях, информация о методиках определения типа пациента по данным цефалометрического анализа, классификация травм и деформаций, а также возможные подходы к синтезу в условиях как врожденных, так и приобретенных деформаций.

Конечной целью работы является выявление наиболее эффективного метода для синтеза 3D-модели послеоперационного облика пациента.

Annotation

Within the limits of the given work the solution of a problem of synthesis of postoperative shape of the patient is considered at traumas and deformations of maxillary department of an obverse skeleton.

As a subject domain necessary to studying, the data about a structure of an obverse skeleton of the person, existing techniques of reception of shape of the patient, the data about cephalometric points, planes and measurements, the information on techniques of definition of type of the patient by data cephalometric the analysis, classification of traumas and deformations, and also possible approaches to synthesis in conditions both congenital, and the got deformations acts.

Work ultimate goal is revealing most the best method for synthesis of 3D-model of postoperative shape of the patient.

Синтез послеоперационного облика пациента — это определенная последовательность операций, применение которой позволяет получить из исходной деформированной (в результате травм или врожденных деформаций) модели лицевого скелета (3D-модель пациента) некоторое приближение к первоначальному облику до появления влияний, вызванных деформациями различного характера (врожденные/приобретенные) и степени тяжести (малая/средняя/высокая).

Процесс синтеза послеоперационного облика пациента можно разделить на два этапа:

1. получение 3D-модели пациента лицевого скелета;
2. изменение геометрии 3D-модели пациента по выбранному методу для получения недеформированного облика, т.е. некоторого эталона, к которому необходимо стремиться при проектировании оперативных вмешательств.

Синтез по 2D фотографиям

Имеются фотографии пациента до травмы.

На примере программы FaceGenModeller [1]. Программа строит полигональную модель головы с мягкими тканями. Редактирование мягкотканной модели можно осуществить по 150 признакам. Данный метод используется в судебной практике.

После получения мягкотканной модели необходима операция преобразования модели в костнотканную.

Объемная съемка

Необходимо наличие специального дополнительного оборудования.

При объемной съемке головы пациента используется следующее оборудование:

- оборудование для получения 3D-данных (объемная съемка);
- оборудование для обработки данных и реконструкции 3D-поверхности для последующего распознавания;
- оборудование для создания биометрического шаблона по полученным чертам лица;
- оборудование для окончательного распознавания облика пациента на основе сравнения полученного и ранее определенного биометрических шаблонов.

Технология трехмерного распознавания лиц

Система трехмерного распознавания лиц, построенная на базе технологии A4Vision, состоит из четырех последовательных ступеней, действий [2]:

Сканер поверхности оцифровывает поверхности объектов, находящихся в поле зрения, и передает их в компьютер для дальнейшей обработки в виде трехмерных сеток.

Программный модуль обработки данных восстанавливает трехмерную модель лица для распознавания.

Программный модуль извлечения антропометрических особенностей лица рассчитывает по полученной модели биометрический шаблон, который в дальнейшем используется в процедуре распознавания.

Программный модуль сравнения используется для сравнения полученного шаблона с зарегистрированными ранее шаблонами из базы данных.

Технология использует «обратный метод Герасимова». Модель на выходе оптимизируется. При использовании в черепно-лицевой хирургии необходимы дополнительные модули, например, определение цефалометрических точек, для возможности работы с эталонными моделями.

Недостатки

- Применение технологии объемной съемки для пациентов с черепно-лицевыми травмами/ деформациями на выходе дает модель с деформациями, а не послеоперационный облик пациента без учета влияния деформаций/травм. Данную технологию можно использовать для получения предоперационной модели пациента или в комбинации с другим методом.
- Применение технологии восстановления облика по 2D-фотографиям невозможно в отсутствие таковых или при наличии значительных дефектов на них. Большое влияние на результат построения модели имеет освещение, при котором сделаны фотографии;
- В результате применения обеих технологий получается только мягкотканная модель головы человека, т.е. чтобы получить костнотканную модель нужно произвести дополнительный расчет. Это повлечет за собой увеличение погрешности.

Метод синтез по томограмме

Модель из STL формата после преобразований (если такие нужны) импортируется в 3Dmax [6].

Построение 3D-модели послеоперационного облика пациента по томограмме

В данном методе учитывается, что начальные данные получаются в результате диагностического исследования пациента с использованием томографа. Используется поддержка основного медицинского формата данных — DICOM. Для этого выбирается программный пакет Amira [3] компании Mercury Systems.

Методика получения 3D-модели пациента и ее импорта в систему трехмерного моделирования и визуализации

Последовательность шагов по построению 3D-модели пациента и ее последующего импорта в систему моделирования и визуализации можно разделить на две части:

1. шаги, производимые в программном обеспечении (ПО) анализа результатов диагностического исследования (Amira);
2. шаги, производимые в ПО трехмерного моделирования и визуализации (3DS MAX) [4].

Преимущества и недостатки метода восстановления по цефалометрическим точкам:

- Модель строится сразу без мягких тканей.
- Цефалометрические точки можно определять на обоих видах модели, что позволит вносить изменения в одной модели и автоматически переносить их на вторую с помощью сравнения цефалометрических измерений.
- Не требуется обратных преобразований (с использованием обратного метода Герасимова), что уменьшает погрешность, так как делается меньше допущений.
- Данный метод не зависит от повреждений.
- Не требует дополнительных исходных данных, т.к. томограммы пациента включены в метод.
- По результатам диагностических исследований можно построить начальный облик и синтезировать требуемый облик пациента, как в виде костно-, так и в виде мягкотканной модели.
- Метод требует базы данных для синтеза недостающих частей изначальной модели.

Алгоритм синтеза по прототипам родственников

Используется как дополнение к методу синтеза по томограммам.

Используются модели родственников. С помощью опорных точек на моделях родственников и имеющихся опорных точках пациента происходит синтез модели пациента. Возможно, синтезировать как мягкотканную модель, так и костнотканную.

Алгоритм использования моделей-прототипов

Данный алгоритм предполагает наличие базы для возможности использования эталонных (без повреждений) моделей. Определяются цефалометрические точки [9] модели пациента. Из базы автоматически выбираются модели, подходящие по антропологическим измерениям [8].

В модели пациента замещается поврежденная часть с соответствующими преобразованиями.

Данный алгоритм применим с методами формирования объемной модели пациента и алгоритмом определения цефалометрических точек.

Алгоритм зеркалирования

Применяется при наличии модели с незначительными локальными повреждениями. Может использоваться как с костнотканной моделью, так и с мягкотканной.

Алгоритм не используется самостоятельно. Применим в комбинации с методами формирования объемной модели.

Литература

1. Официальный сайт FaceGenModeller: <http://www.facegen.com/>;
2. Журнал CCTV Фокус, «Технология трехмерного распознавания лиц»: <http://www.procctv.ru/page.php?id=450>;
3. официальный сайт компании Mercury Systems (Amira): <http://www.tgs.com>;
4. официальный сайт компании Autodesk (3DS MAX): <http://www.autodesk.com>;
5. официальный сайт компании Medicim (Medical Image Computing): <http://medicim.com/>;
6. документация по программе Amira: <http://www.amiravis.com/usersguide41/usersguide/index.html>;
<http://www.amiravis.com/amira41-PDFindex.html>;
<http://www.amiravis.com/usersguide41/usersguide/faq.html>;
7. Российский портал развития, «Augmented ID – опознание проводит коммуникатор»: http://www.russia-gateway.org/content/NEWS/NewsItem_3718612/
8. Краниометрия. Методика антропологических исследований, В.П. Алексеев, Г.Ф. Дебец;
9. Справочник по стоматологии, под ред. А.И. Рыбакова и Г.М. Иващенко, 1977;
10. Совершенствование оказания медицинской помощи на этапах лечения и реабилитации пострадавших с травмами челюстно-лицевой области, Д.К. Калиновский, И.Н. Матрос-Таранец, С.Б. Алексеев, Т.Н. Хახелева, Донецкий государственный медицинский университет им. М.Горького, □Донецк, Украина, 2006;

НЕЙРОСЕТЕВАЯ ОБРАБОТКА ДАННЫХ РАМАНОВСКОЙ СПЕКТРОСКОПИИ

Ивченко М.О.

Научный руководитель: Канд. техн. наук, доцент, Власов А.И.

МГТУ им. Н.Э. Баумана, кафедра Иу4, Москва, Россия

NEURAL NETWORK PROCESSING OF RAMAN SPECTROSCOPY DATA

Ivchenko M.O.

Ph.D., associate professor, Vlasov A.I.

MSTU n.a. N.E. Bauman, Moscow, Russia

Аннотация

В данной работе рассматривается использование искусственных нейронных сетей с обратным распространением ошибок в качестве альтернативы традиционным методам PLS и PCR для идентификации компонент в веществе по его спектру рамановского рассеяния. Автором предложена архитектура такой сети, а также рассмотрен метод вейвлет-фильтрации для очищения спектров от искажений, вызванных шумами или флуоресценцией.

Annotation

This paper describes the use of back-propagation neural networks as an alternative to traditional PLS and PCR methods for identifying components in a substance by its Raman scattering spectrum. The author proposes an architecture for such a network and describes a wavelet filtering method for ridding spectra of distortions caused by noise and fluorescence.

Рамановская спектроскопия – неупругое рассеяние оптического излучения на молекулах вещества, как твёрдого, жидкого, так и газообразного, сопровождающееся заметным изменением его частоты. В спектре рассеянного излучения появляются спектральные линии, число и расположение которых определяется молекулярным строением вещества.

Спектры рамановского рассеяния уникальны для каждого вещества, являясь, тем самым, его «отпечатком пальца». Это обстоятельство, наряду с малым временем, необходимым для получения спектра, которое может колебаться от секунд до нескольких минут, позволяет осуществлять быстрый и чувствительный спектральный анализ образца. Список аналитических применений рамановской спектроскопии продолжает расширяться по сей день; одно из важнейших применений заключается в определении молекулярного состава образца по его спектру рамановского рассеяния.

Для идентификации отдельных компонент в веществах по их спектру традиционно используют методы многомерного анализа данных. К ним относятся метод наименьших квадратов (Partial Least Squares – PLS) и регрессия на главных компонентах (Principal Component Regression – PCR) [1].

Однако распознаванию спектров рамановского рассеяния мешает ряд проблем. Присутствие флуоресцентных соединений, примесей и сложных смесей добавляют трудностей при идентификации соединений по их спектрам. Более того, концентрация исследуемого компонента бывает так мала, что она находится вблизи границы предела обнаружения используемого измерительного инструмента.

В связи с вышесказанным, перспективным представляется создание адаптивной системы обнаружения и идентификации компонентов соединений, которая могла бы работать в условиях повышенной зашумленности входных спектров рамановского рассеяния.

В данной работе в качестве такой системы рассматривается искусственная нейронная сеть (ИНС) с обратным распространением ошибки как возможная альтернатива методам PLS и PCR.

Создание и обучение ИНС

ИНС представляют собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Нейронные сети не программируются в привычном смысле этого слова, они *обучаются*. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами (таких, как PLS и PCR). Технически обучение заключается в нахождении коэффициентов связей между нейронами.

В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что, в случае успешного обучения, сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искаженных данных, что идеально подходит для решения поставленной в работе задачи [2]. С целью упрощения системы перспективным является применение многопроходной процедуры идентификации, в которой для поиска каждого компонента из библиотеки строится и обучается отдельная ИНС.

Входными данными для каждой из обучаемых ИНС (рисунок 1) является n -канальный спектр рамановского рассеяния отдельных компонентов, которые могут присутствовать в образцах, исследуемых при эксплуатации системы.

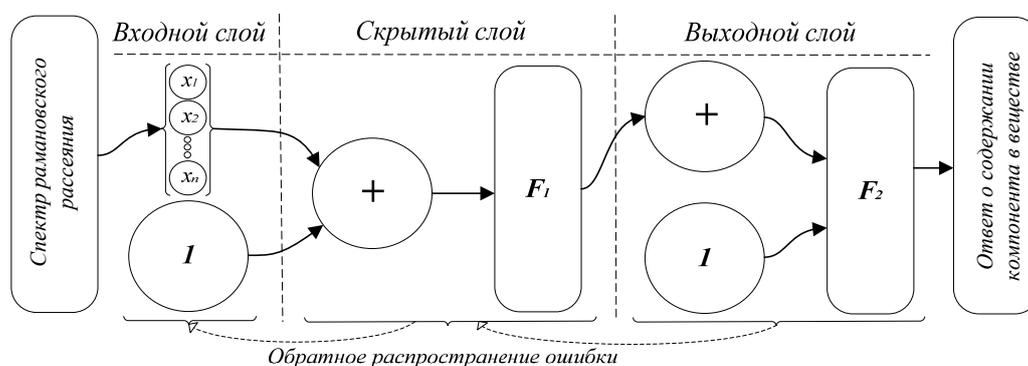


Рис.1. ИНС для выявления содержания компонента в веществе по его спектру рамановского рассеяния.

Для нейронов скрытых слоев целесообразно использовать сигмоидальную активационную функцию $F(x) = (1 + e^{-gx})^{-1}$ с коэффициентом усиления $g = 1$, а для входного и выходного слоев — линейную $F(x) = x$. Однако выбор активационных функций, равно как и числа нейронов скрытого слоя, необходимо варьировать при обучении ИНС с тем, чтобы обеспечить наиболее корректную работу сети.

Также необходимо учесть, что в алгоритме обратного распространения ошибки добавление нейронного смещения позволяет сдвигать начало отсчета сигмоидальной активационной функции, и приводит к ускорению процесса обучения. Эта возможность может быть легко введена в обучающий алгоритм с помощью добавляемого к каждому нейрону веса. Этот вес обучается так же, как и все остальные веса, за исключением того, что подаваемый на него сигнал всегда равен +1, а не выходу нейрона предыдущего слоя [3].

Подготовка спектров к распознаванию

Анализ как микроскопических, так и макроскопических образцов с помощью рамановского спектрографа часто приводит к получению спектров, искаженных шумами или флуоресценцией. Ранее автором в [5] были представлены два метода цифровой обработки спектров рамановского рассеяния: цифровая фильтрация на основании преобразования Фурье и сглаживание. В данной работе представлен другой перспективный способ очищения

спектров от шума: *вейвлет-фильтрация*. Как показано в [4], этот метод позволяет извлечь количественную информацию из слабых и зашумленных спектров рамановского рассеяния.

Для применения вейвлет-фильтрации к спектру рамановского рассеяния его необходимо представить в виде сигнала $f(t)$ на конечном интервале времени, причем за t принять волновое число спектра. Вейвлет-алгоритм разбивает входной сигнал на различные частотные компоненты. В этом он похож на традиционное преобразование Фурье, однако вейвлет-алгоритм использует функции, локализованные в пространстве с нулевым средним значениям (*вейвлеты*) вместо обычных синусоидальных, что позволяет иметь доступ к информации как о частотной, так и о временной зависимостях.

Сигнал $f(t)$ представляется в виде суммы элементарных вейвлетов и декомпозируется на два сигнала. Один из них содержит низкочастотные компоненты (аппроксимация A_1), а второй – флуктуации сигнала (детальный сигнал D_1). Алгоритм итеративно применяется к аппроксимированной части функции и называется дискретным вейвлет-преобразованием (ДВП). Сигнал может быть восстановлен с помощью алгоритма обратного дискретного вейвлет-преобразования (ОДВП).

Если не включать последний аппроксимирующий компонент A_n в процесс ОДВП, то более гладкая часть сигнала не будет входить в сгенерированный спектр. В случае рамановских спектров этот фоновый компонент в основном вызывается рассеянием излучения и флуоресценцией. В то же время, исключая первые компоненты декомпозиции, можно добиться удаления ненужного шума в спектре. На рисунке 2 показан пример вейвлет-фильтрации рамановского спектра. Декомпозиция проводилась до $8^{\text{-го}}$ уровня (D_1, \dots, D_8, A_8), в то время как восстановление спектра выполнялось с использованием компонент D_5-D_8 , тем самым удаляя шум и фоновые сигналы [4].

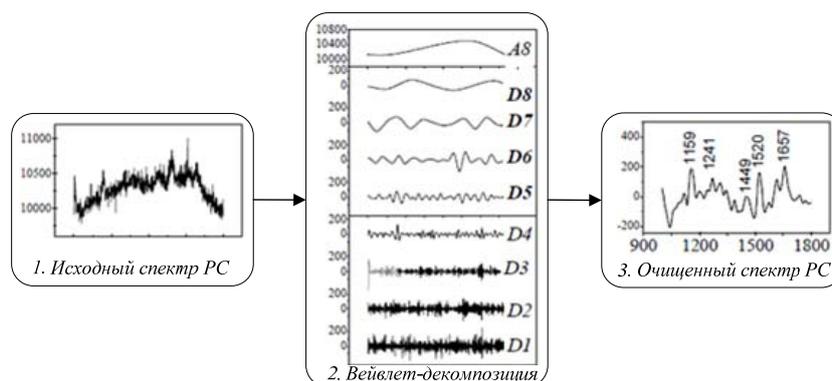


Рис.2. Очищение спектра рамановского рассеяния от шумов вейвлет-фильтрацией [4].

Следует отметить, что этот метод не требует какой-либо подготовки данных и является автоматическим численным алгоритмом, подходящим для внедрения в аналитические системы для пользователей с ограниченными знаниями математического аппарата.

Нейросетевое распознавание спектров рамановского рассеяния

Алгоритм распознавания спектров рамановского рассеяния представлен на рисунке 3. Сверху представлены исследуемые данные, а снизу – процессы обработки этих данных. Исходный n -канальный спектр рамановского рассеяния (1) подвергается вейвлет-фильтрации (2). Очищенный от шума спектр (3) поступает на вход обученных ИНС обратного распространения (4), каждая из которых определяет присутствие одного из компонентов в исследуемом спектре. Каждый компонент имеет соответствующую ему ИНС. Данный процесс подвергается распараллеливанию при условии наличия аппаратной поддержки платформы, на которой разворачивается система. На основании выходов ИНС формируется аналитический отчет о составе исследуемого вещества (5).

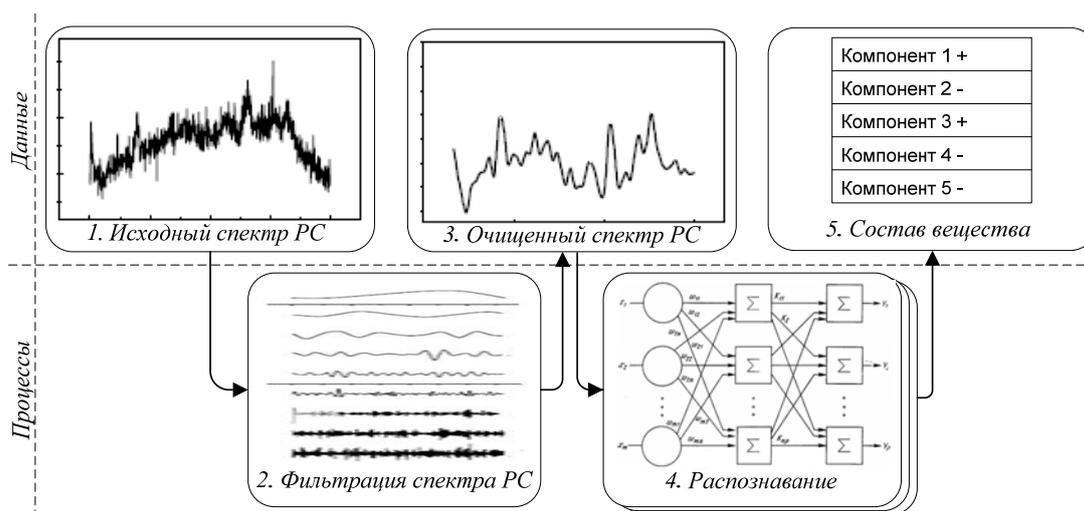


Рис.3. Алгоритм интеллектуального распознавания спектров рамановского рассеяния

Представляемый отчет позволяет идентифицировать компоненты, входящие в состав вещества, исследуемого с помощью рамановской спектроскопии.

Выводы

Использование ИНС с обратным распространением ошибок имеет ряд преимуществ над традиционными методами PLS и PCR.

Во-первых, способность сети к обобщению дает возможность изначально обучить ее корректно обрабатывать наихудший сценарий, когда рамановский спектр сильно зашумлен. Затем на вход можно подавать как идеальные, так и искаженные сигналы, а сеть, в виду способности к обобщению, будет выдавать правильный результат. Во-вторых, обученная ИНС обрабатывает данные, связанные нелинейным отношением, лучше, чем традиционные линейные методы. Наконец, ИНС в виду своей архитектуры постоянно совершенствуется, подстраиваясь под различные виды и уровни зашумленности входного сигнала, обеспечивая, таким образом, адаптивность системы.

Однако ИНС не лишено проблем. Прежде всего, нет гарантии, что сеть может быть обучена за конечное время. Много усилий, израсходованных на обучение, пропадает напрасно после затрат большого количества машинного времени. Когда это происходит, попытка обучения повторяется без всякой уверенности, что результат окажется лучше. Нет также уверенности, что сеть обучится наилучшим возможным образом. Алгоритм обучения может попасть в «ловушку» так называемого локального минимума и будет получено худшее решение.

Не возникает сомнений, что при условии качественного обучения представленный в работе подход является одним из наиболее перспективных для решения поставленной задачи.

Литература

1. Shaver J.M. Chemometrics for Raman Spectroscopy. Handbook of Raman Spectroscopy // Practical Spectroscopy Series Vol. 28, Marcel Dekker Inc. New York. 2001. 28-29.
2. Искусственная нейронная сеть: http://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть.
3. Vanderbeek R. Back-Propagation Network for Analog Signal Separation in High Noise Environments // U.S. Army Armanent Munitions Chemical Command. Maryland. 1992.
4. Camerlingo C. et al. Wavelet data processing of micro-Raman spectra of biological samples // Measurement Science and Technology Vol. 17. United Kingdom. 2006. 298-303.

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ БЫСТРОГО РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ

Кузьмин М.Б.

*Научный руководитель к.т.н., доцент Федорук В.Г.
МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Россия*

NEURAL NETWORKS FOR RAPID FACE RECOGNITION

Mikhail B. Kuzmin

*Supervisor Ph.D., Associate professor Vladimir G. Fedoruk
Moscow State Technical University n.a. N.E. Bauman*

Аннотация

В данной статье рассмотрена возможность использования нейронных сетей с алгоритмом обучения на основе самоорганизации для быстрого распознавания изображений. В тексте приведены основные принципы, положенные в основу алгоритма быстрого поиска, а также кратко рассмотрен эксперимент на пробном наборе изображений и изложены полученные результаты. Сделаны выводы о возможности использования алгоритма.

Abstract

This article considers the possibility of using self-organizing neural networks for rapid image recognition. The text contains the basic principles of the rapid search algorithm, as well as the description of the experiment on a trial set of images and presents the results of the experiment. At the end of the article there is a conclusion about the possibility of using the algorithm and probable modifications.

Распознавание изображений – одна из наиболее широких и перспективных областей применения нейронных сетей. Была исследована возможность использования нейронных сетей для быстрого поиска изображений лица человека, похожих на заданное, в большой базе данных (порядка 10^6 людей). Для идентификации используются ключевые признаки, полученные в результате обработки изображения с помощью алгоритма PCA или преобразований Фурье.

При обработке изображений создается база данных, в которой для идентификации каждого человека используется заданное количество изображения (в исследуемом случае 6). Основным концептуальным подходом для быстрого поиска была выбрана кластеризация пространства критериев с целью построения иерархической структуры классов. Впоследствии поиск изображений будет значительно ускорен за счет локализации области поиска в пределах определенного класса.

Требования к алгоритму кластеризации.

Поставленная задача включает в себя набор правил, при невыполнении которых дальнейшее использование базы и поиск будет невозможным. Такими правилами являются:

1. Все изображения одного лица должны быть отнесены к одному классу.
2. Классы изображений не должны пересекаться – каждое изображение может входить только в один класс.
3. Структура классов является иерархической – от больших классов, включающих большое количество изображений до малых, содержащих всего несколько.
4. Поиск должен осуществляться быстро и однозначно.

Ранее задача поиска была решена последовательным перебором всех векторов признаков с вычислением евклидовой меры для поиска наиболее похожего. Однако это требует больших временных затрат.

Концептуальная схема алгоритма.

На начальном этапе работы алгоритма осуществляется обработка изображений и получение векторного пространства признаков каждого изображения. Они представляют собой набор векторов, где каждый вектор характеризует конкретное изображение из базы данных.

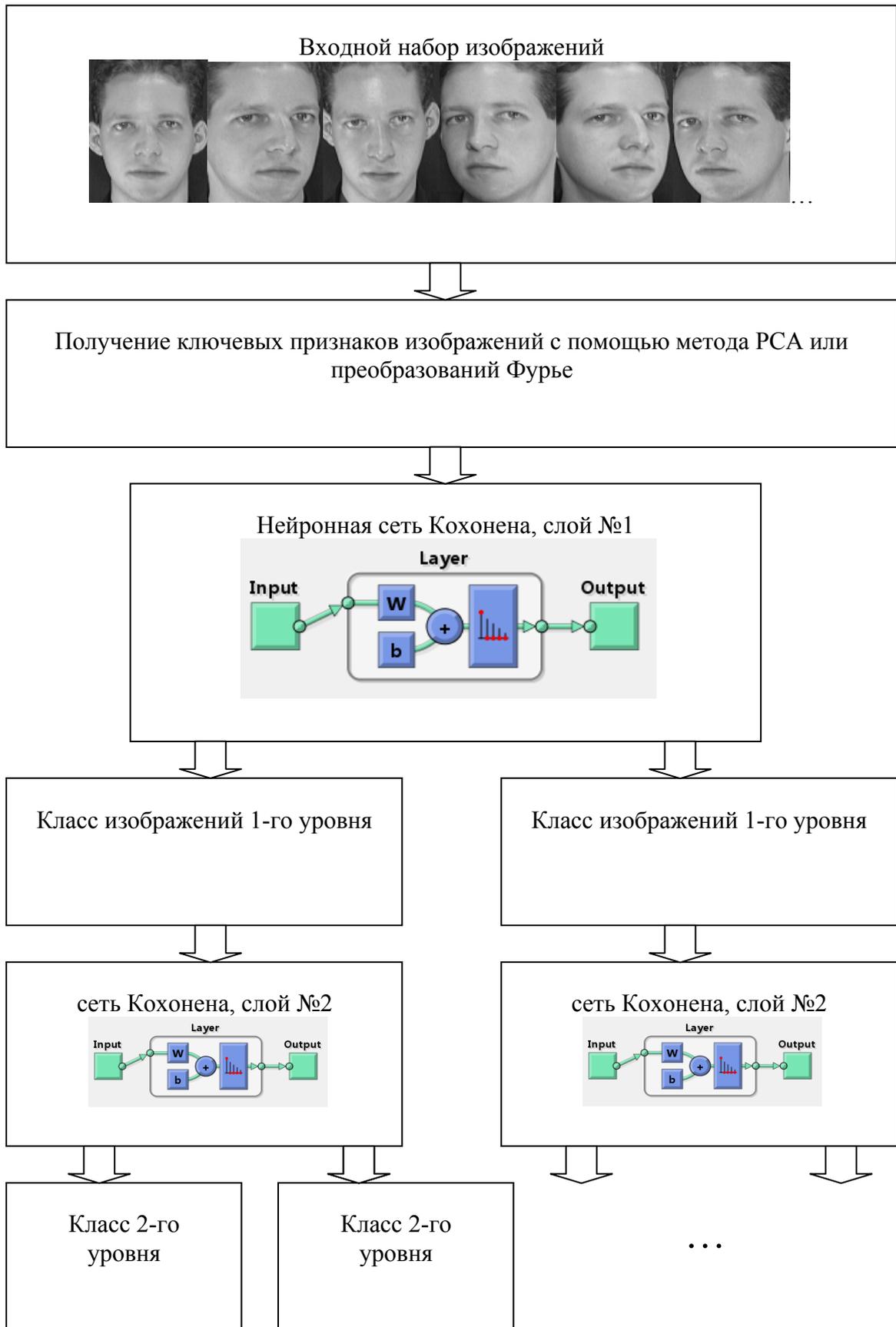


Рис.1. Концептуальная схема алгоритма.

Затем полученный набор векторов приводится к виду, подходящему для обучения нейронной сети Кохонена. Параметры сети зависят от входных данных и требуемого количества классов. На входы всех нейронов последовательно подаются вектора признаков и в процессе обучения выделяются отдельные классы. Затем каждый из полученных классов снова подвергается процедуре обработки. В результате получают классы второго, третьего и так далее уровней, вплоть до выделения отдельных изображений.

Особенности кластеризации изображений.

Задача разбиения изображений на классы осложнена тем, что в базе присутствуют снимки одного лица с разных ракурсов и с разным освещением. Кроме того, в базе могут быть различные варианты внешности одного и того же человека – например в очках и без них.

Краткое описание экспериментов.

Для исследования возможности применения нейронных сетей на основе самоорганизации для быстрого поиска изображений в базе данных использовался пакет MATLAB. В нем была реализована сеть с обучением по алгоритму Кохонена, адаптированная для результатов обработки изображений. В качестве базы изображений был выбран пробный набор из 240 снимков, по 6 на каждое лицо.

Параметры нейронной сети Кохонена для решения данной задачи сильно зависят от входных данных и требований к результатам кластеризации. Было реализовано несколько пробных сетей, на основе результатов их обучения были сделаны выводы о возможности использования такого алгоритма в дальнейшем.

Для получения признаков изображений использовались алгоритм PCA и преобразования Фурье. В результате обработки изображений с помощью PCA для каждого изображения был получен вектор из 56 признаков. Для кластеризации данных применялась сеть из 40 нейронов с 56 входами и коэффициентом обучения 0,1. Такое количество нейронов было задано для проверки возможности выделения в отдельный класс 6 изображений одного лица. Из-за большой амплитуды входных данных количество циклов обучения было задано 50000 итераций.

В процессе экспериментов изменялось:

1. количество используемых признаков – для уменьшения пространства критериев;
2. количество нейронов – для исследования оптимального размера класса.

После обучения сети на вход подавался тот же набор входных данных и проводился анализ результатов:

1. Количество однозначно определенных лиц: 7 – все изображения для данных лиц были отнесены к одному классу. Пример таких изображений представлен на рис.2.



Рис.2. Пример однозначно определенных изображений.

2. Количество частично определенных лиц: 28 – больше половины изображений было отнесено к одному классу. Пример таких изображений представлен на рис 3.



Рис.3. Пример неоднозначно определенных изображений.

3. Количество неопределенных лиц : 12 – меньше половины изображений было отнесено к одному классу. Пример таких изображений:



Рис.4. Пример неопределенных изображений.

Выводы.

В результате анализа возможности использования нейронной сети Кохонена для быстрого распознавания изображений были получены неоднозначные результаты. С одной стороны, для некоторых изображений (в наилучшем случае около 25%) достигается однозначно определение с возможностью последующего быстрого поиска, для большого количества изображений (около 75%) вероятность распознавания составляет больше 50%.

Также, очевидно, что использование характеристик изображений, полученных с помощью метода главных компонент(РСА) приоритетнее из-за лучших результатов выделения отдельных лиц. При этом, уменьшение размерности используемых признаков нецелесообразно из-за ухудшения результатов распознавания.

Однако видно, что существуют изображения, которые не могут быть однозначно определены и распознаны в качестве одного объекта при обработке, как методом главных компонент, так и с помощью преобразований Фурье. Возможно, развитие алгоритма для улучшения работы с такими изображениями в направлении комбинирования обоих наборов признаков или поэтапного фильтрования хорошо определенных изображений.

Литература

1. Г.А. Кухарев «Биометрические системы» Спб. «Политехника»: 2001 г. – 240с
2. Осовский С. «Нейронные сети для обработки информации» М.: 2002 г. – 344с
3. Саймон Хайкин Нейронные сети: полный курс. 2-е издание Хайкин С., "Вильямс", 2006 г., 1104 с.

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ МЕТОДА РОЯ ЧАСТИЦ В ЗАДАЧЕ ПРИБЛИЖЕННОГО ПОСТРОЕНИЯ МНОЖЕСТВА ПАРЕТО

Антух А.Э., Семенихин А.С., Хасанова Р.В.

д.ф.-м.н., проф. Карпенко А.П.

МГТУ им. Н.Э. Баумана, кафедра РКБ, Москва, Россия.

INVESTIGATION OF PARTICLE SWARM OPTIMIZATION EFFICIENCY IN APPROXIMATE CONSTRUCTION OF THE PARETO SET

A. Antukh, A. Semenikhin, R. Khasanova

professor A. Karpenko.

MSTU n.a. Bauman, Moscow, Russia.

Аннотация

Рассматривается метод роя частиц для приближенного построения множества Парето в задаче многокритериальной оптимизации. Приведено формальное описание задачи и алгоритма, реализующего приближенное построение множества Парето. На одной тестовой задаче выполнено исследование эффективности алгоритма.

Abstract

This paper discusses an implementation of particle swarm optimization method for Pareto set definition in a multi criteria problem. The paper contents a definition of multi criteria problem and description of the particle swarm optimization method for Pareto set definition. Also the paper provides a report about effectively research for described method using test problem.

Введение

В настоящее время все более широкое распространение получают стохастические поведенческие методы оптимизации [1]. Одним из таких методов является метод роя частиц (*Particle Swarm Optimization*, PSO), основанный на закономерностях социального поведения, например, поведения птиц в стае, косяков рыб и т.п. [2], [3]. В данном методе индивидам ставятся в соответствие частицы в пространстве параметров задачи оптимизации.

Достаточно новым является применение PSO-метода в задаче многокритериальной оптимизации (*Multi-Objective Swarm Optimization*, MOPSO). В работе рассматривается применение этого метода для приближенного построения множества Парето в указанной задаче.

Не формально, множество Парето может быть определено как множество, в котором значение любого из частных критериев оптимальности можно улучшить только за счет ухудшения других частных критериев – любое из решений, принадлежащее множеству Парето, не может быть улучшено одновременно по всем частным критериям.

1. PSO-метод

Обозначим множество частиц $\mathbf{P} = \{P_i, i \in [1 : N]\}$, где N – количество частиц в рое (размер популяции). В момент времени $t = 0, 1, 2, \dots$ координаты частицы P_i определяются вектором $X_{i,t} = (x_{i,t,1}, x_{i,t,2}, \dots, x_{i,t,n})$, а ее скорость – вектором $V_{i,t} = (v_{i,t,1}, v_{i,t,2}, \dots, v_{i,t,n})$, где n – размерность вектора варьируемых параметров. Начальные координаты и скорости частицы P_i равны $X_{i,0} = X_i^0$, $V_{i,0} = V_i^0$ соответственно.

Итерации в каноническом методе PSO выполняются по схеме

$$V_{i,t+1} = \alpha V_{i,t} + U[0, \beta] \otimes (X_{i,t}^b - X_{i,t}) + U[0, \gamma] \otimes (X_{g,t} - X_{i,t}),$$
$$X_{i,t+1} = X_{i,t} + V_{i,t+1}.$$

Здесь $U[a,b]$ представляет собой n -мерный вектор псевдослучайных чисел, равномерно распределенных в интервале $[a,b]$; \otimes - символ покомпонентного умножения векторов; $X_{i,t}^b$ - вектор координат частицы P_i с наилучшим значением целевой функции $\Phi(X)$ за все время поиска; $X_{g,t}$ - вектор координат соседней с данной частицы с наилучшим за время поиска значением целевой функции $\Phi(X)$; α, β, γ - свободные параметры алгоритма.

Важной частью метода роя частиц для задач многокритериальной оптимизации является определение глобально лучшей частицы для каждой частицы популяции. В однокритериальном случае глобально лучшая частица определялась выбором частицы с наилучшей позицией. В многокритериальной задаче глобально лучшую частицу (*best local guide*) следует искать на множестве Парето.

В работе использована одна из модификаций канонического PSO-метода – метод Хью и Эберхарта (метод «меняющихся» соседей) применительно к двумерной двухкритериальной задаче оптимизации [4]. В этом случае поиск глобально лучшей частицы для каждой частицы популяции осуществляется в пространстве критериев двухкритериальной задачи следующим образом: первое расстояние между частицами вычисляется на основе первого частного критерия оптимальности. Таким образом, для каждой частицы находим k ее соседей. Затем, используя второй частный критерий оптимальности, находим лучшую частицу для частицы P_i из этих соседей. Это и будет глобально лучшая частица для частицы P_i .

У метода Хью и Эберхарта есть недостаток: выбор глобально лучшей частицы очень сильно зависит от выбора критерия, которого нужно считать фиксированным [3].

2. Исследование эффективности метода

Исследование эффективности MOPSO-метода выполнено для следующей двумерной двухкритериальной задачи:

$$\phi_1(X) = x_1^2 + x_2^2, \quad \phi_2(X) = (x_1 - 1)^2 + (x_2 - 1)^2; \quad (1)$$

$$D_X = \{X \mid 0 \leq x_i \leq 1, i = 1, 2\}. \quad (2)$$

Здесь $\phi_1(X)$, $\phi_2(X)$ - частные критерии оптимальности, D_X множество допустимых значений вектора варьируемых параметров $X = (x_1, x_2)$. Известно, что для указанной задачи множество Парето D_X^* представляет собой отрезок прямой, соединяющий точки (0, 0) и (1, 1) [1].

На основе значительного количества экспериментов было установлено, что оптимальными для данной задачи являются следующие параметры: размер популяции $P_s = 100$; количество итераций $N = 500$. Некоторые результаты экспериментов иллюстрирует рисунки 1а), 1б), на которых, соответственно, представлены приближенные фронт Парето и множество Парето задачи (1), (2).

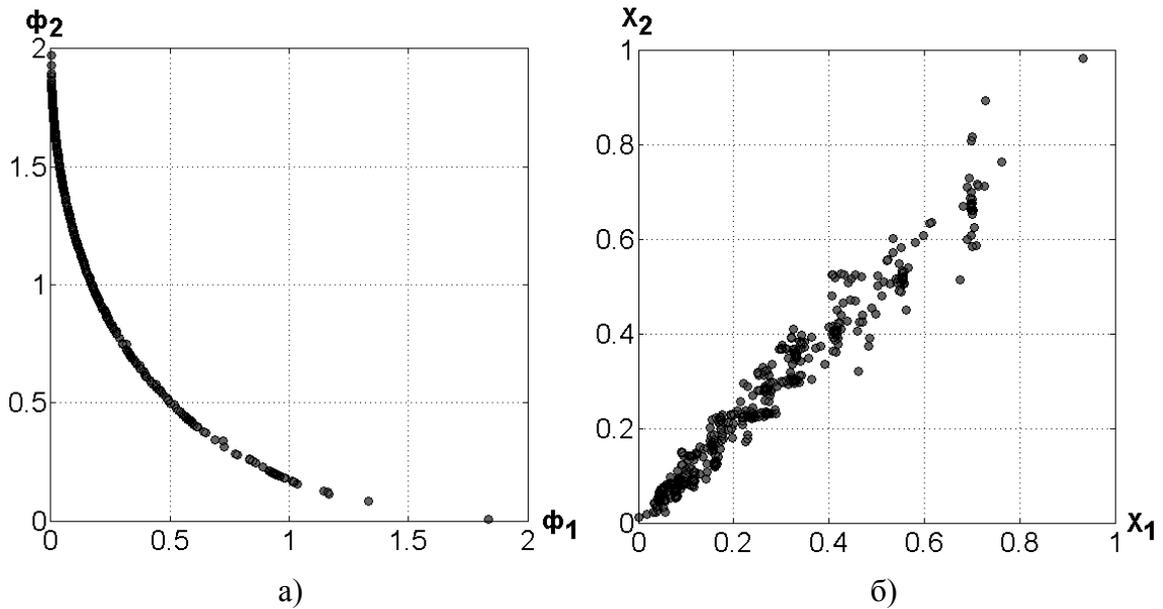


Рис. 1. Приближенные фронт и множество Парето задачи (1)

Как видно из рисунка 1а), количество точек во множестве Парето значительно превышает размер популяции. Это связано с тем, что реализованный метод использует специальный архив для сохранения точек свободных от доминирования. Архив обновляется по мере появления новых недоминируемых точек.

Получение результатов, представленных на рисунке 1, потребовало примерно 10 минут расчетов на персональном компьютере с процессором 2,16 Гц и 2 ГБ оперативной памяти.

Необходимо отметить, что при решении задачи (1) полным перебором на достаточно точной сетке, требуемое процессорное время для простых частных критериев оптимальности сравнимо с указанным выше временем. Однако, при увеличении сложности критериев, MOPSO-метод позволяет найти решение приемлемой точности за существенно меньшее время.

Поскольку известно точное решение задачи (1), имеется возможность оценить эффективность MOPSO-метода не только визуально, но и численно. Для этого было вычислено среднее отклонение частицы от идеального множества (отрезка прямой). Рисунок 2 иллюстрирует полученные результаты.

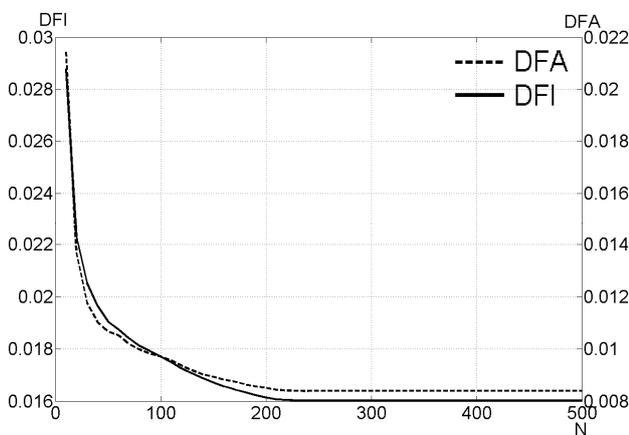


Рис. 2. Отклонение от точного решения

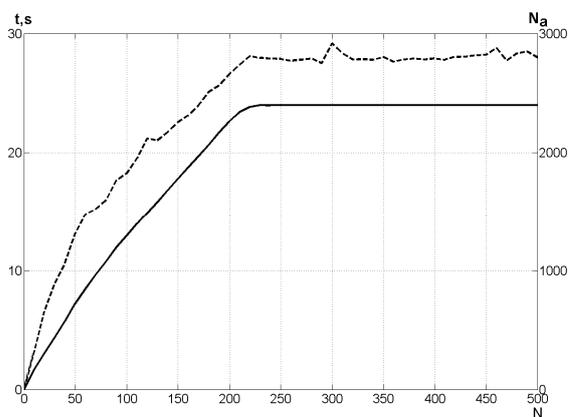


Рис. 3. Время расчета и размер обрабатываемого архива

Сплошной линией на рисунке 2 показано среднее отклонение от точного решения (DFI), как функция числа итераций (N). Рисунок показывает, что метод сходится уже на первых 100-200 итерациях, затем отклонение выходит на некоторый постоянный уровень. На последних 300 итерациях значения не изменяются, но DFI не отражает форму вычисленного фронта Парето, поэтому нельзя говорить о стагнации решения без дополнительных исследований.

Пунктиром на рисунке 2 показана величина среднего отклонения всех частиц (DFA) от DFI (среднего отклонения от точного решения). Данная величина в некоторой степени демонстрирует тот факт, что в архивном множестве Парето нет частиц, которые расположены суперблизко и супердалеко от точного решения, т.е. полученное множество точек распределено по области в некоторой степени равномерно.

Дополнительно, было выполнено исследование производительности алгоритма. Для этого было выявлено, каким образом с течением итераций растет размер архива и как это влияет на общее время решения задачи. Некоторые результаты экспериментов приведены на рисунке 3. Пунктиром на рисунке показано время, затрачиваемое на расчет каждой 10 итераций; сплошная линия показывает размер архива (N_A) в функции количества итераций (N). Заметим, что в данном варианте алгоритма был использован архив, не имеющий ограничения на его размер. Это позволило накапливать в архиве все Парето оптимальные точки. Для реальных задач многокритериальной оптимизации необходимо ограничивать размер архива, чтобы его неограниченный рост не привел к нехватке памяти компьютера.

Заключение. Работа демонстрирует подход к приближенному построению множества Парето к задаче многокритериальной оптимизации с помощью модифицированного метода роя частиц. Проведенное исследование показало, что данный метод, будучи относительно простым математически и в реализации обеспечивает, решение задачи с заданной точностью.

К недостаткам метода можно отнести то, что выбор глобально лучшей частицы очень сильно зависит от выбора критерия, которого нужно считать фиксированным. Для преодоления этого недостатка можно использовать модифицированные методы, приведенные, например, в статье [5].

В развитие работы планируется также реализация критерия останова метода, основанного на отсутствии роста размера архива в течение заданного количества итераций.

Литература

1. Карпенко А.П. Методы оптимизации [Электронный ресурс] / (<http://bigor.bmstu.ru>).
2. Карпенко А.П., Селиверстов Е.Ю. Глобальная оптимизация методом роя частиц. Обзор // Информационные технологии, 2010, № 2, с. 25-34.
3. Субботин С.А., Олейник Ан.А., Олейник Ал.А. PSO-метод, Интеллектуальные мультиагентные методы (Swarm Intelligence) [Электронный ресурс] / (<http://www.csit.narod.ru/>).
4. Hu X., Eberhart R. Multiobjective optimization using dynamic neighborhood particle swarm optimization // World Congress on Computational Intelligence: Proceeding, 2002.- pp. 1677–1681.
5. Mostaghim S., Teich, J. Strategies for Finding Good Local Guides in Multi-Objective Particle Swarm Optimization (MOPSO) // Swarm Intelligence Symposium: Proceeding, 2003.- pp. 26–33.

МНОГОКРИТЕРИАЛЬНАЯ ОПТИМИЗАЦИЯ НА ОСНОВЕ НЕЧЕТКОЙ АППРОКСИМАЦИИ ФУНКЦИИ ПРЕДПОЧТЕНИЙ ЛИЦА, ПРИНИМАЮЩЕГО РЕШЕНИЯ

Д.А. Моор, Д.Т. Мухлисуллина

д.ф.-м. н., профессор Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РК6 (САПР), Москва, Российская Федерация

MULTI-CRITERIA OPTIMIZATION BASED ON FUZZY APPROXIMATION OF THE PREFERENCES FUNCTION OF A DECISION MAKER

Moor D.A., Mukhlisullina D. T.

Professor, Karpenko A.P.

CAD/CAM/CAE dep., Bauman Moscow State Technical University, Moscow, Russian Federation

Аннотация

Рассматривается прямой адаптивный метод многокритериальной оптимизации на основе аппроксимации функции предпочтения лица, принимающего решение, с помощью аппарата нечеткой логики. Исследуется эффективность метода при решении 2-х и 3-х критериальных тестовых задач с использованием различных сверток критериев оптимальности.

Abstract

The paper considers the direct adaptive method for multi-criteria optimization based on approximation of a decision maker preferences function by using the fuzzy logic apparatus. Research results of the method effectiveness for solving 2-criterial and 3-criterial tests problems with different criteria convolutions are presented.

1. Постановка задачи

Пусть X - вектор варьируемых параметров задачи. Множеством допустимых значений вектора X является ограниченное и замкнутое множество D_X . Лицо, принимающее решение (ЛПР), стремится минимизировать на множестве D_X каждый из частных критериев оптимальности $\phi_1(X), \phi_2(X), \dots, \phi_m(X)$, что условно записывается в виде

$$\min \Phi(X) = \Phi(X^*), X \in D_X, \quad (1)$$

где X^* - искомое решение задачи многокритериальной оптимизации (МКО-задачи). Предполагается, что частные критерии оптимальности тем или иным способом нормализованы.

Обозначим $\varphi(X, \Lambda)$ операцию свертки частных критериев оптимальности, где $\Lambda \in D_\Lambda \subset R^m$ - вектор весовых множителей; $D_\Lambda = \{\lambda_i | \lambda_i \geq 0, \sum_i \lambda_i = 1, i \in [1:m]\}$ - множество допустимых значений этого вектора. Для любого $\Lambda \in D_\Lambda$ метод скалярной свертки сводит решение задачи (1) к решению однокритериальной задачи глобальной условной оптимизации (ОКО-задачи)

$$\min \varphi(X, \Lambda) = \varphi(X^*, \Lambda), X \in D_X. \quad (2)$$

Условие (2) позволяет полагать [1], что функция предпочтений ЛПР определена не на множестве D_X , а на множестве D_Λ : $\psi: \Lambda \rightarrow R$. В результате МКО-задача сводится к задаче выбора вектора $\Lambda^* \in D_\Lambda$ такого, что $\max_{\Lambda \in D_\Lambda} \psi(\Lambda) = \psi(\Lambda^*)$, $\Lambda \in D_\Lambda$. При этом вместо функции $\varphi(X, \Lambda)$ используется дискретная функция $\psi(\Lambda)$ [2], так что вместо указанной задачи решается задача

$$\max \psi(\Lambda) = \psi(\Lambda^*) = \psi^*, \Lambda \in D_\Lambda. \quad (3)$$

2. Метод решения задачи

Общая схема рассматриваемого прямого адаптивного метода решения МКО-задачи является итерационной и состоит из перечисленных ниже следующих основных этапов [3].

Этап «разгона» метода. МКО-система некоторым образом (например, случайно) последовательно генерирует k векторов $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ и для каждого из этих векторов Λ_i выполняет следующие действия. 1) Решает соответствующую ОКО-задачу. 2) Предъявляет ЛПР найденное решение X_i^* , а также соответствующие значения всех частных критериев оптимальности. 3) ЛПР оценивает эти данные и вводит в МКО-систему соответствующее значение своей функции предпочтений $\psi(\Lambda_i)$.

Первый этап. На основе всех имеющихся в МКО-системе значений $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ вектора Λ и соответствующих оценок функции предпочтений $\psi(\Lambda_1), \psi(\Lambda_2), \dots, \psi(\Lambda_k)$ МКО-система выполняет следующие действия. 1) Строит функцию $\tilde{\psi}_1(\Lambda)$, аппроксимирующую функцию $\psi(\Lambda)$ в окрестности точек $\Lambda_1, \Lambda_2, \dots, \Lambda_k$. 2) Отыскивает максимум функции $\tilde{\psi}_1(\Lambda)$ – решает ОКО-задачу $\max \tilde{\psi}_1(\Lambda) = \tilde{\psi}_1(\Lambda^*) \quad \Lambda \in D_\Lambda$. 3) С найденным вектором Λ_1^* решает ОКО-задачу вида (2) – находит вектор параметров и соответствующие значения частных критериев оптимальности, а затем предъявляет их ЛПР. ЛПР оценивает указанные данные и вводит в систему соответствующее значение своей функции предпочтений $\psi(\Lambda_1^*)$.

Второй этап. На основе всех имеющихся в системе значений $\Lambda_1, \Lambda_2, \dots, \Lambda_k, \Lambda_1^*$ вектора Λ и соответствующих оценок функции предпочтений $\psi(\Lambda_1), \psi(\Lambda_2), \dots, \psi(\Lambda_k), \psi(\Lambda_1^*)$ МКО-система выполняет аппроксимацию функции $\psi(\Lambda)$ в окрестности точек $\Lambda_1, \Lambda_2, \dots, \Lambda_k, \Lambda_1^*$ – строит функцию $\tilde{\psi}_2(\Lambda)$ и т.д. по схеме первого этапа до тех пор, пока ЛПР не примет решение о прекращении вычислений.

3. Алгоритм и программная реализация метода

Используем в качестве свертки (2) аддитивную свертку

$$\varphi(X, \Lambda) = \sum_i \lambda_i \phi_i(X), \quad i \in [1:m].$$

Входами системы нечеткого вывода являются значения весов частных критериев оптимальности – нечеткие термы $\lambda_{i,j,k}$, $i, j=1, 2, \dots, k \in [1:m]$. Выходной переменной системы нечеткого вывода является лингвистическая переменная ψ , ядро которой $\tilde{\psi}$ принимает значения $1, 2, \dots, 9$ [4].

Взаимосвязь между входными и выходными переменными описывается нечеткими правилами вида:

ЕСЛИ <значения входных переменных> ТО <значение выходной переменной> (f_i).

Здесь $f_i \in [0,1]$ – коэффициенты определенности (веса нечетких правил).

Используется схема нечеткого вывода Мамдани, который выполняется за два шага.

Шаг 1. Положим, что выполнено N экспериментов по определению значений лингвистической переменной ψ . Пусть в n_1 этих экспериментов переменная ψ приняла значение ψ_1 , в n_2 экспериментах – значение ψ_2 и т.д. до n_9 и ψ_9 . Соответствующие входные векторы Λ обозначим $\Lambda_{i,j} = (\lambda_{i,j,1}, \lambda_{i,j,2}, \dots, \lambda_{i,j,m}) \in D_\Lambda$, где $i \in [1:9]$, $j \in [1:n_i]$.

Матрицу знаний $\{\lambda_{i,j,k}, i \in [1:9], j \in [1:n_i], k \in [1:m]\}$ можно представить в виде

$$\bigcup_{j=1}^{n_i} \left[\bigcap_{k=1}^m (\lambda_j = \lambda_{i,j,k}) \right] \rightarrow \psi_i, \quad i \in [1,9]$$

Шаг 2. Тонкая настройка модели $\psi(\Lambda)$ (параметрическая идентификация модели) осуществляется путем подбора следующих параметров: полуширина a функций принадлежности входных переменных $\mu_{i,j,k}^a$; полуширина b функций принадлежности выходных переменных $\mu_{i,r,k}^b$; величина δ , определяющая закон изменения весовых множителей правил. Область допустимых значений указанных параметров представляет собой параллелепипед D_P , а критерий оптимальности имеет вид

$$E(a, b, \delta) = \sqrt{\sum (\psi(\Lambda_i) - \bar{\psi}(\Lambda_i))^2}, \quad i \in [1: N],$$

где $\bar{\psi}(\Lambda_i)$ - результат нечеткого логического вывода Мамдани в точке Λ_i .

Таким образом, формально задача параметрической идентификации формулируется в виде следующей задачи глобальной многомерной условной оптимизации [3]

$$\min E(a, b, \delta) = E(a^*, b^*, \delta^*) \quad (a, b, \delta) \in D_P. \quad (4)$$

4. Исследование эффективности метода

Исследование выполнено для двух двумерных двухкритериальных задач и одной двумерной трехкритериальной тестовых МКО-задач:

- двухкритериальная задача 1, имеющая выпуклый фронт Парето,
 $\phi_1(X) = (x_1 - 5)^2 + x_2^2 + 10$; $\phi_2(X) = x_1^2 + (x_2 - 5)^2 + 20$; $D_X = \{X \mid 0 \leq x_i \leq 5, i = 1,2\}$;
- двухкритериальная задача 2 (невыпуклый многосвязный фронт Парето)

$$\phi_1(X) = -x_1; \quad \phi_2(X) = -f(X) \left(1 - \sqrt{\frac{x_1}{f(X)}} - \frac{x_1}{f(X)} \sin(10\pi x_1) \right);$$

$$f(X) = 1 + 9x_2; \quad D_X = \{X \mid 0 \leq x_i \leq 1, i = 1,2\};$$

- трехкритериальная задача 3 (выпуклый фронт Парето)

$$\phi_1(X) = (x_1 - 5)^2 + x_2^2 + x_3^2 + 10; \quad \phi_2(X) = x_1^2 + (x_2 - 5)^2 + x_3^2 + 20;$$

$$\phi_3(X) = x_1^2 + x_2^2 + (x_3 - 5)^2 + 5; \quad D_X = \{X \mid 0 \leq x_i \leq 5, i = 1,2,3\}.$$

Исследование выполнено для $k = 3, 5, 7$ «разгонных» значений вектора Λ .

При $k = 3$ «разгонных» точках решение МКО-задачи 1 потребовало 12 итераций. Вид функции предпочтений ЛПР $\psi(\Lambda_1)$ после 12 итераций представляет рис. 1а). Здесь и далее в число итераций включаются и «разгонные» итерации; светлые кружки соответствуют значениям функции предпочтений ЛПР на промежуточных итерациях, темный кружок – на последней итерации. При $k = 5$ «разгонных» точках решение задачи потребовало 16 итераций, при $k = 7$ - 11 итераций.

Для МКО-задачи 2 при $k = 3$ вид функции $\psi(\Lambda_2)$ после 14 итераций представлен на рис. 2б). При $k = 5$ «разгонных» точках решение задачи потребовало 14 итераций, при $k = 7$ - 14 итераций.

Вид линий уровня функции $\psi(\Lambda_3)$ после 17 итераций ($k = 3$) решения МКО-задачи 3 показывает рис. 2. При $k = 5$ «разгонных» точках решение задачи потребовало 11 итераций, при $k = 7$ - 16 итераций.

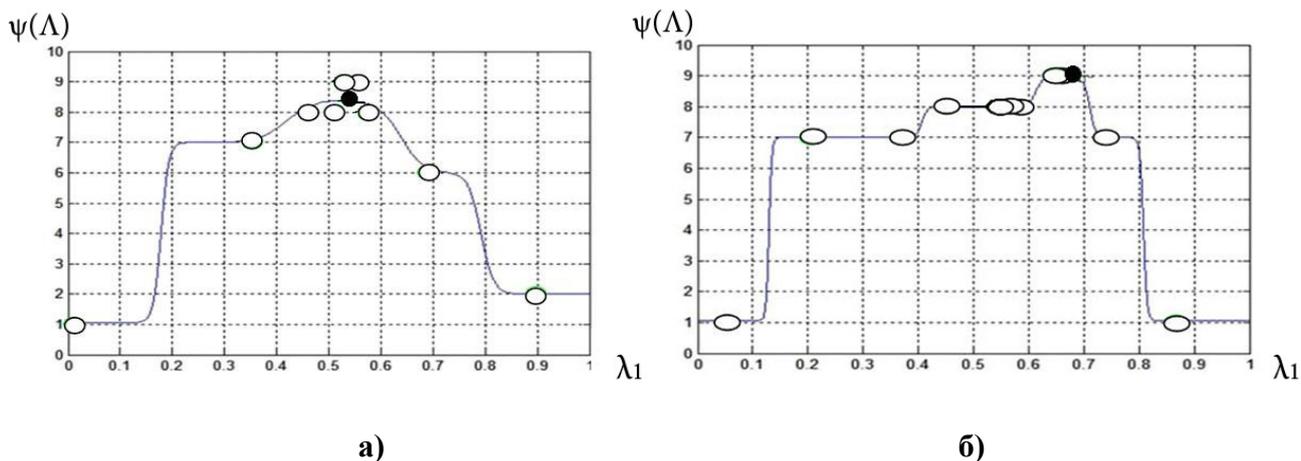


Рис. 1. Аппроксимация функция предпочтений ЛПР:
 а) МКО-задача 1; $k = 3$; 12-я итерация; б) МКО-задача 2; $k = 3$; 14-я итерация

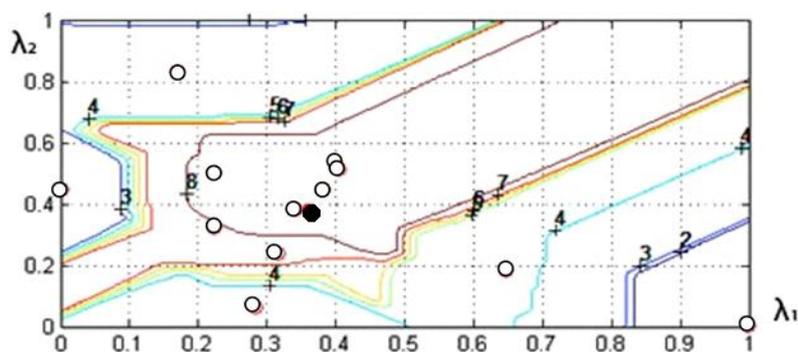


Рис. 2. Линии уровня функции, аппроксимирующей функцию предпочтений ЛПР:
 МКО-задача 3; $k = 3$; 17-я итерация

К недостаткам метода можно отнести то, что с каждой следующей итерацией вычислительная сложность пути Фаззификация \rightarrow Агрегирование \rightarrow Активизация \rightarrow Аккумуляция \rightarrow Дефаззификация повышается, поскольку увеличивается количество входных правил нечеткой системы.

Также было проведено исследование эффективности использования сверток критериев оптимальности, отличных от линейной аддитивной свертки:

- 1) свертка Гермейера $\varphi_G(X, \Lambda) = \min \frac{\phi_i(X)}{\lambda_i}, i \in [1 : m]$;
- 2) свертка на основе отклонения от идеальной точки $\varphi_\rho(X, X^{**}) = \rho(X, X^{**})$, где X^{**} - идеальная точка, ρ - некоторая метрика.

Заметим, что в работе в качестве метрики ρ использовалась евклидова метрика, хотя можно использовать и другие метрики, например, чебышевскую метрику $\rho(X, X^{**}, \Lambda) = \max \lambda_i |x_i - x_i^{**}|, i \in [1 : m]$.

Результаты экспериментов подтвердили, что не все точки множества Парето могут быть получены с помощью линейной аддитивной свертки (например, если множество Парето не выпукло или несвязно). Напротив, свертка Гермейера позволяет находить все точки множества Парето, включая слабо эффективные точки. Если ЛПР интересуют только точки, принадлежащие множеству Парето, это свойство свертки Гермейера следует считать ее недостатком. Недостатком свертки на основе идеальной точки является произвол в выборе метрики.

Литература

1. Лотов, А.В. Введение в экономико-математическое моделирование / А.В.Лотов.– М.: Наука, 1984.- 392 с.
2. Карпенко, А.П. Один класс прямых адаптивных методов многокритериальной оптимизации / А.П. Карпенко, В.Г. Федорук // Информационные технологии.- 2009.- №5.- С.24-30.
3. Карпенко, А.П. Нейросетевая аппроксимация функции предпочтений лица, принимающего решения, в задаче многокритериальной оптимизации / А.П.Карпенко, Д.Т. Мухлисуллина, В.А. Овчинников // Информационные технологии.- 2010.- (в печати).
4. Ротштейн, А.П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети / А.П. Ротштейн.- Винница: УНИВЕРСУМ-Винница, 1999.- 320 с.

КОНЦЕПЦИЯ АДАПТИВНОЙ ИММУННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Плехов С. Ю.
МГТУ им.Н.Э.Баумана, Москва

FUNDAMENTAL OF IMMUNE ADAPTIVE IT SYSTEM

Plekhov S. Y.
MSTU named after Bauman, Moscow

Аннотация

В статье перечислены основные проблемы больших вычислительных комплексов и предложен подход, позволяющий решить все основные проблемы заказчиков и производителей с использованием адаптивной иммунной информационной системы, отличающейся способностью приспосабливаться к изменившимся условиям, в числе которых: аппаратные отказы, изменения конфигурации и рекомендованные к выполнению действия, с контролем выполняемых действий со стороны производителя и заказчика.

Abstract

The article lists the main problems of large computing systems and proposed an approach that allows to solve all basic problems of customers and manufacturers with an adaptive immune system information other than the ability to adapt to changing conditions, including: hardware failures, configuration changes and recommended to step in control actions performed by the manufacturer and the customer.

Мы являемся свидетелями бурного роста технологий отработки информации, и вместе с этим, растет необходимость не только в высокопроизводительном оборудовании, но также немаловажным фактором является эффективность, непрерывность и отказоустойчивость данного процесса. В последнее время на рынке IT появилось множество коммерческих продуктов, позволяющих управлять, проводить мониторинг в реальном времени, диагностировать проблемы и узкие места, однако, данными средствами довольно тяжело повысить эффективность и на это есть целый ряд причин. Перед каждым конкретным заказчиком стоит нелегкая задача по накоплению и, что еще более важно, сохранению накопленных знаний о комплексе, полученных в процессе эксплуатации. Обычно такой компетенцией владеют несколько человек, с уходом которых процесс накопления знаний приходится начинать фактически с нуля. Конечно, на рынке существуют различные системы управления знаниями, но пользование такими решениями зачастую требует немало сил и ощутимо снижает эффективность работы персонала, но является единственным выходом сохранить компетенцию для конкретного заказчика. Вендору также непросто доносить знания до заказчиков. Помимо обязательного минимума знаний, который персонал может получить, окончив курсы и пройдя сертификацию вендора, существует ряд проблем, которые требует от заказчика незамедлительных действий. Например, в случае, если обнаружена критическая ошибка в микрокоде СХД, которая может привести к потере критических данных, необходимо немедленно запланировать работы, чтобы нейтрализовать такую угрозу, но такая информация рассылается в виде бюллетеня с рекомендациями, либо даются директивы сервис - центру рекомендовать всем заказчикам обновления, но несмотря на все усилия, существует процент заказчиков, до которых такая информация попадает с непозволительным опозданием.

Некоторые комплексы могут быть настолько сложными, что даже при наличии максимально подробной инструкции тот или иной заказчик будет не в состоянии ее выполнить. Так, в случае реализации многоступенчатой инструкции, которую необходимо адаптировать под каждого заказчика, специалистами могут быть допущены множественные ошибки, и в случае если все они не будут найдены, ПО или аппаратное решение будет

пущено в промышленную эксплуатацию в конфигурации, не соответствующей рекомендованной поставщиком и последствия такой эксплуатации могут быть катастрофическими. Самой важной причиной является отсутствие у клиентов необходимой компетенции для проектирования, реализации и использования сложных вычислительных комплексов. Обычной практикой в таких случаях является обращение к системным интеграторам либо к вендорам, что зачастую не является выходом из положения. Системный интегратор, не заинтересован в конечности начатых проектов, а ценообразование на основе фиксированного процента относительно стоимости конечного продукта заставляет предлагать заказчикам дорогие, и как следствие, громоздкие решения, в некоторых конфигурациях несовместимыми между собой, результатом чего является затягивание фазы внедрения, повышение конечной стоимости проекта, отсутствие возможности сопровождать громоздкие продукты своими силами и заказчик попадает под зависимость от конкретного интегратора.

В случае обращения к вендору, упор делается на платформозависимые решения, призванные оставить заказчика в числе клиентов данного вендора. Мы подошли к тому, что необходимо ПО, которое снизит нагрузку на человека в процессе ввода в эксплуатацию комплекса, автоматизирует процесс мониторинга, реагирования на события и, что вероятно, самое главное - обеспечить развитие всего комплекса. Не так давно появился новый тип продуктов, позволяющий накапливать знания, к примеру, программные комплексы, занимающиеся обеспечением ПО, которые управляют установками приложений и ОС, а также всех необходимых параметров, что при правильном использовании позволяет накапливать и использовать накопленные знания при обеспечении рабочих мест.

В последнее время становятся популярны концепции иммунных информационных систем и, действительно, такой подход кажется довольно интересным. Такие системы способны держать в равновесии информационную систему, реагируя описанным способом на predetermined перечень событий, в числе которых аппаратные отказы, сбои программного обеспечения и пр. Однако, стоит отметить, что данный подход хоть и является новаторским, все же имеет ряд серьезных минусов. Так, сопровождение не включает в себя такие важные этапы как переход на следующие версии ПО, диагностирование проблем с низкой производительностью оборудования, необходимость замены его на более мощное и пр. Количество описываемых вариантов воздействий и реакций на эти воздействия в идеале стремится в бесконечность, как следствие, этот способ не подходит для динамично развивающихся систем. Даже с возможностями к обучению, данная система не сможет сравниться по компетенции с вендорами или крупными заказчиками. Использование иммунных информационных систем является качественным улучшением, но решает далеко не все проблемы. Однако, если при таком подходе использовать нейросети с функционалом, достаточным для принятия решений, а контроль и оценку конкретных решений, принимаемых данным комплексом, можно возложить на квалифицированного оператора. Чем больше сеть, тем лучшие результаты на выходе можно будет получить. Обучение и накопление знаний система должна получить от квалифицированного оператора, который должен принять корректные решения при нештатных ситуациях. Такой подход позволит обмениваться опытом между заказчиками, объединяя нейросети, несмотря на то, что оборудование и конфигурация у заказчиков может быть различна.

Следующим этапом должен быть единый модульный мультивендорный продукт, поддерживаемый основными производителями ПО и аппаратных комплексов, позволяющий не только накапливать знания, но и делиться ими. Данный процесс должен контролироваться вендорами, системой принятия решений, и, непосредственно заказчиком.

В случае, если удастся добиться интереса к такой системе от вендора, у которого аккумулируются все знания по его продуктам, то такое сотрудничество решит большинство проблем эксплуатации больших информационных комплексов. Получая отчеты о критичных проблемах от заказчиков, даже не объединенных в общую нейросеть, осуществляющую управление и принятие решений, вендор сможет оперативно ставить вопрос об обновлений перед заказчиком, решающих критическую проблемы, и обучать систему корректно обращаться со сложными программными или аппаратными комплексами. Таким образом, можно существенно облегчить повседневную работу специалистов, а заказчиков избавить от необходимости в высококвалифицированных специалистах, соответственно понизить стоимость эксплуатации всего комплекса.

Литература

1. M. Burgess: Knowledge Management and Promises. Lecture Notes in Computer Science 2009 ;Volum 5637. s. 95-107
2. Alva L. Couch, Hengky Susanto, Marc Chiarini: Modeling Change Without Breaking Promises. AIMS 2007: 97-108
3. Kyrre M. Begnum, Mark Burgess, John A. Sechrest: Adaptive provisioning using virtual machines and autonomous role-based management. ICAS 2006: 7

БЫСТРОЕ ПОСТРОЕНИЕ МНОЖЕСТВА ДОСТИЖИМОСТИ ВЫСОКОМАНЕВРЕННОГО ЛЕТАТЕЛЬНОГО АППАРАТА

Козлова О.Г., Федин В.А.

д.ф.-м.н., профессор Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РК6, Москва, Россия

MANEUVERABLE AIRCRAFT ATTAINABILITY SET FAST CONSTRUCTION

V.A. Fedin, O.G. Kozlova

professor A.P. Karpenko

BMSTU, Moscow, Russia

Аннотация

Рассматривается задача быстрого построения области достижимости динамической системы. Исследуется подход к решению этой задачи на основе многократного интегрирования модельной системы обыкновенных дифференциальных уравнений при различных управлениях и нейросетевой метод.

Abstract

Dynamic system attainability domain fast construction problem is discussed. We analyze this problem solution approach based on multiple model differential equations system integration with different controls values, and neuronet method.

1. Постановка задачи

Уравнения движения центра масс летательного аппарата (ЛА) в нормальной земной системе координат $OXYZ$ описываются следующей системой нелинейных дифференциальных уравнений [1]:

$$\begin{cases} \dot{V} = g \cdot (n_T - \sin \Theta); \\ \dot{\Theta} = \frac{g}{V} (n \cdot \cos \gamma_c - \cos \Theta); \\ \dot{\Psi} = -g \cdot n \cdot \sin \gamma_c / V \cdot \cos \Theta; \\ \dot{X} = V \cdot \cos \Theta \cdot \cos \Psi; \\ \dot{Y} = V \cdot \sin \Theta; \\ \dot{Z} = -V \cdot \cos \Theta \cdot \sin \Psi. \end{cases} \quad (1)$$

где V – скорость ЛА, Θ – угол наклона траектории, Ψ – угол поворота траектории, Y – высота ЛА, n_T – тангенциальная перегрузка, n – нормальная перегрузка, γ_c – скоростной угол крена, g – ускорение свободного падения.

Управления ЛА задаются вектором

$$u^T = (n_T, n, \gamma_c); |n_T| \leq n_T^{\max}; |n| \leq n^{\max}; |\gamma_c| \leq \pi.$$

Область достижимости ЛА определяется как множество значений вектора координат его центра масс в момент времени T , полученных при заданном начальном состоянии ЛА и всех возможных допустимых управлениях. Очевидно, что динамика области достижимости может быть описана динамикой ее границ.

В работе [1] были получены структуры управлений, приводящих на границы области достижимости при фиксированном угле крена. Структура управления, приводящего на дальнюю границу области достижимости, имеет вид (рисунок 1):

$$\mathbf{u}_\delta = \begin{bmatrix} u_1 = \pm 1 \\ u_2 = 1 \end{bmatrix}_{t \in [0, t^-]} \cup \begin{bmatrix} u_1 = 0 \\ u_2 = 1 \end{bmatrix}_{t \in [t^-, T]}, \quad (2)$$

где t^- – время перехода на особый участок управления $n_{\text{особ}} = 0$. Здесь $n^{\text{max}} \cdot u_1 = n$, $n_T^{\text{max}} \cdot u_2 = n_T$, $\mathbf{u}_s = [u_1, u_2]^T$, $|u_1| \leq 1$, $|u_2| \leq 1$.

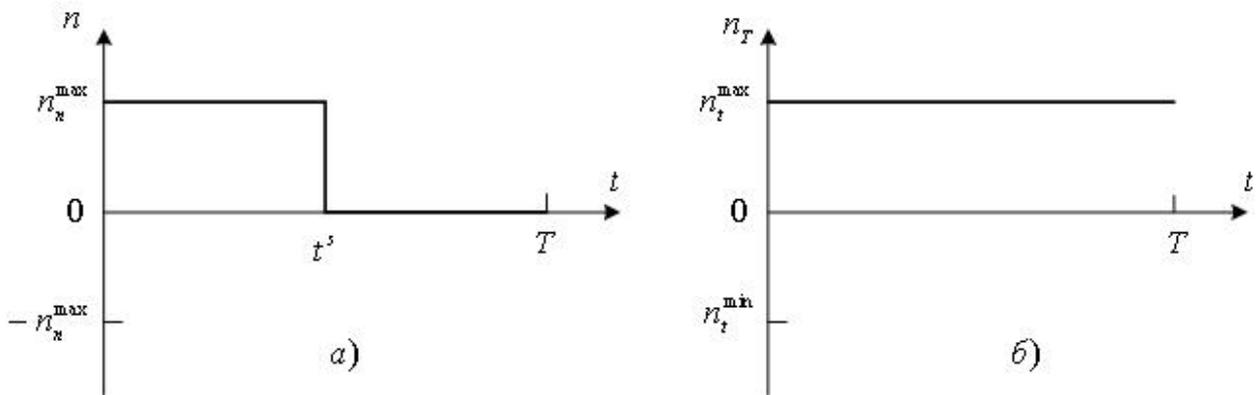


Рис. 1. Структура управлений, формирующих дальнюю границу области достижимости

Путем многократного интегрирования системы (1) при различных управлениях (2) (т.е. при варьировании времени переключения t и угла крена) может быть построена дискретная аппроксимация границы Γ_γ области достижимости [2].

2. Параллельное построение границы области достижимости

Схема метода мультифиниша и декомпозиции управлений по процессорам подробно описана в работе [2].

Для экспериментального исследования эффективности метода мультифиниша разработана MPI-программа, реализующая этот метод.

Эксперименты выполнены на виртуальном кластере, созданном с помощью программной системы VMware и функционирующим под управлением свободно распространяемой операционной системы Ubuntu. Результаты экспериментов иллюстрирует рисунок 2, который показывает, что на числе процессоров от 2 до 10 достигается ускорение, близкое к расчетному.

3. Нейросетевой метод

Была разработана нейронная сеть, определяющая координаты границы области достижимости ЛА при задании граничного управления. Входной слой сети состоит из элементов V_0 , Θ_0 , T , γ , t , где $0 \leq t \leq T$ – момент времени переключения нормальной перегрузки n со значения n^{max} до 0, $-\pi \leq \gamma \leq \pi$ – угол крена ЛА. Сеть содержит два скрытых слоя, содержащих одинаковое количество персептронов с функцией активации – гиперболический тангенс. Выходной слой сети состоит из трех нейронов с линейной функцией активации, соответствующих координатам X, Y и Z конечного положения ЛА.

Рассматривается задача (1), в которой входные величины варьируются в следующих пределах:

$$1.6 \cdot V_{3g} \leq V_0 \leq 1.9 \cdot V_{3g}, \quad -\frac{\pi}{8} \leq \Theta_0 \leq \frac{\pi}{8}, \quad 3c \leq T \leq 5c.$$

Здесь $V_{3g} = 331.46 \text{ м/с}$ – скорость звука в воздухе [3].

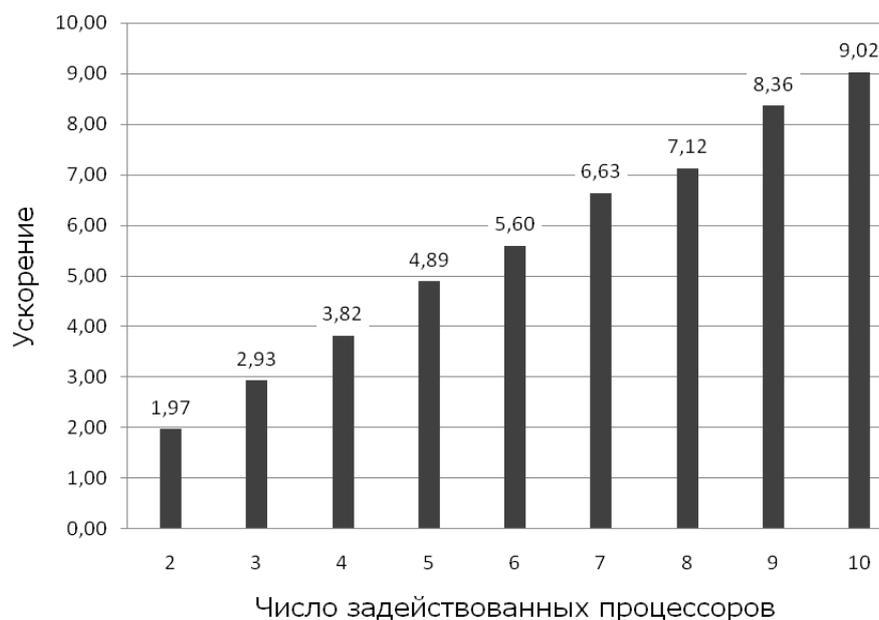


Рис. 2. Экспериментальные результаты (метод мультифиниша)

На рисунке 3 приведены результаты исследования точности аппроксимации дальней границы области достижимости нейронными сетями с различным количеством нейронов. Рисунок показывает, что с увеличением числа нейронов в сети погрешность аппроксимации стабильно уменьшается. При увеличении числа нейронов с 22 до 40 средняя погрешность уменьшилась в 2.5 раза.

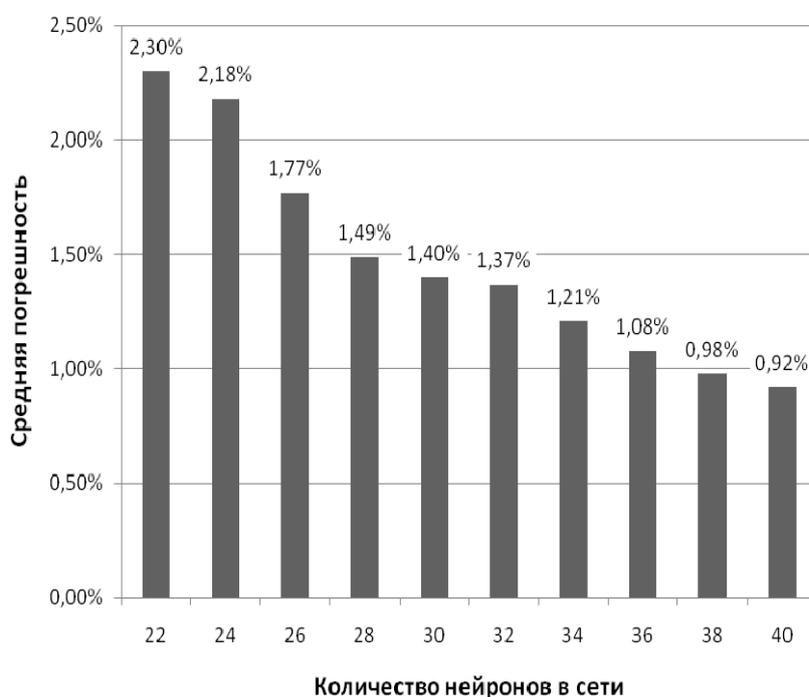


Рис. 3. Средняя погрешность аппроксимации дальней границы области достижимости нейронными сетями с различным количеством нейронов

Пример аппроксимации дальней границы области достижимости, полученной с помощью нейронной сети, приведен на рисунке 4.

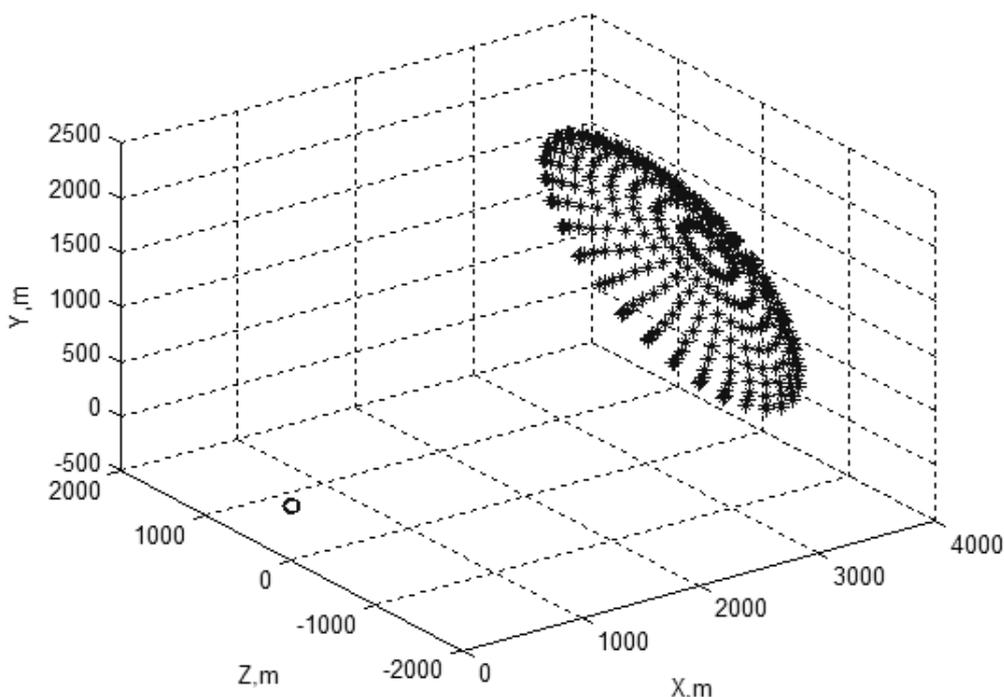


Рис. 4. Аппроксимация нейронной сетью дальней граница области достижимости, построенной при начальных условиях $V_0 = 1.9 \cdot V_{36}$, $\Theta_0 = \pi/8$, $T = 5$ с.

Заключение

В работе рассмотрены два метода быстрого построения области достижимости динамической системы на примере высокоманевренного летательного аппарата. По итогам исследования оба метода можно признать эффективными и заслуживающими более глубокого изучения с привлечением усложненной модели летательного аппарата. Полученные выводы верны также и для построения ближней и боковой границ области достижимости летательного аппарата [3].

Дальнейшая работа над исследованием эффективности параллельного построения области достижимости ЛА ведется на кластере МГУ SKIF.

Планируется провести сравнительное исследование достоинств и недостатков каждого из методов.

Авторы благодарят Воронова Е.М. за постановку задачи и плодотворное обсуждение результатов, Трофимова А.Г. – за концепцию подхода к нейросетевой аппроксимации области достижимости, Вишневецкого К.О. – за помощь в построении границы области достижимости методом мультифиниша.

Литература

1. Воронов Е.М., Карпунин А.А. Алгоритм оценки границ области достижимости летательного аппарата с учетом тяги // Вестник МГТУ. Сер. Приборостроение.- 2007.- №4(69).- С. 81-99.
2. Воронов Е.М., Карпенко А.П., Козлова О.Г., Федин В.А. Численные методы построения области достижимости динамической системы // Наука и образование: электронное научно-техническое издание, www.techomag.edu.ru, январь, 2010.
3. Козлова О.Г. Нейросетевая аппроксимация границы области достижимости летательного аппарата. 1. Трехмерный случай // Наука и образование: электронное научно-техническое издание, www.techomag.edu.ru, август, 2009.

РАЗРАБОТКА ПРИНЦИПОВ СЕРВИС-ОРИЕНТИРОВАННОЙ АРХИТЕКТУРЫ, ОСНОВАННОЙ НА СОБЫТИЯХ

Бочарников А.А., Макарова А.Е.

Жук Д.М., к.т.н., доцент РК6

МГТУ им. Н. Э. Баумана, кафедра РК6, г. Москва

DEVELOPMENT OF MAIN PRINCIPLES OF SEMANTIC EVENT-BASED SERVICE-ORIENTED ARCHITECTURE

Bocharnikov A., Makarova A.

Guk D., PhD, professor of Robotics Dept.

Student of Robotics Dept., BMSTU, Moscow

Аннотация

Нередко звучат высказывания о том, что сервис-ориентированная архитектура (SOA – Service Oriented Architecture) является гарантированным решением: для систем интеграции прикладных программ предприятия (в единую корпоративную систему), для реализации межорганизационных бизнес-процессов, и даже как основное решение для разработки всех интернет-ориентированных (web-based) приложений. Однако, SOA, без семантического описания данных, процедур и моделей передачи информации неспособна обеспечить гибкую и динамическую инфраструктуру. Далее в этой статье мы опишем, как именно семантическая информация должна использоваться в SOA. А так же представим основные принципы семантической среды выполнения, которая объединяет сервис-ориентированную архитектуру и архитектуру, основанную на событиях.

Abstract

Service-Oriented Architecture (SOA) is commonly lauded as a silver bullet for Enterprise Application Integration, inter-organizational business processes implementation, and even as a general solution for the development of all complex Web-oriented applications. However, SOA without semantic descriptions of its data, processes and messaging models fails to achieve a truly flexible and dynamic infrastructure. In this paper we explain where semantics are necessary for SOA and present early work on a Semantic Execution Environment which couples the Service-Oriented and Event-Driven architecture styles with formal semantics.

Введение

Сервис-ориентированная архитектура (SOA) сейчас воспринимается, как следующий шаг в развитии архитектуры программного обеспечения. При помощи SOA можно создавать более сложные программные комплексы при более низкой цене, по сравнению с обычным подходом создания программного обеспечения. Хотя использование WSDL, XML и BPEL позволяет решать широкий круг задач, проблемы начинают быстро появляться, когда необходимо интегрировать сервисы, использующие различные способы представления данных, описания функций, которые они предоставляют. В дальнейшем, функции, предоставляемые сервисом, будем называть процедурами. Семантическое описание всех составляющих веб-сервиса, а именно услуги, предоставляемые сервисом, его данные и процедуры, является важным шагом на пути к созданию более гибкой сервис-ориентированной архитектуры. Под семантическим описанием данных подразумевается семантическое описание различных структур данных (например, классов), которыми оперирует сервис. Для использования семантических веб-сервисов (SWS – Semantic Web Service) необходимо наличие специальной, семантической среды выполнения (CCB, SEE – Semantic Execution Environment), в которой возможно осуществлять поиск, выбор, объединение и вызов сервисов на основе семантической информации. Написание спецификации для такой среды выполнения возложено на OASIS Semantic Execution Environment (SEE).

Существующие решения

Основой SOA являются разбиение сложных информационных систем на четко структурированные, независимые, вызываемые, распределенные, крупнозернистые составляющие, т.е. сервисы. Основным требованием SOA является отделение спецификации

сервиса от его реализации. Благодаря этому достигается высокая степень интеграции компонентов системы, что является основным техническим преимуществом SOA. Несмотря на успех SOA до сих пор остаются не решенными некоторые проблемы. Поиск подходящих сервисов обычно является частью любого процесса разработки программы, основанной на SOA, но нет никакого метода эффективной поддержки поиска требуемого сервиса. Наиболее известным реестром описания сервисов в интернете, основанным на XML, является UDDI. Недостатком UDDI является то, что описание сервиса является чисто синтаксическим, его содержание (смысл) недоступно пользователю и остается на его усмотрение. Аналогичным образом происходит построение сервисов, т.е. оно ведется только на синтаксическом описании, предоставляемым WSDL. Такого рода описание является необходимым, но недостаточным, так как оно не содержит семантической информации, а именно семантические данные остаются неявными и не могут быть автоматически обработаны.

Дополнительные архитектурные неудобства появляются вследствие того, что в обычной реализации SOA клиенты устанавливают синхронное, туннельное соединение с сервером. В первую очередь необходимо отметить, что синхронные соединения не всегда подходят, а иногда и вовсе невозможны, так как они могут привести к большой латентности и понижению качества обслуживания. Во-вторых, требуется дальнейшее уменьшение связанности сервисов. Перед тем, как клиент инициирует запуск сервиса, ему необходимо заранее знать, используется сервис другим клиентом или нет. Появление нового сервиса в существующей SOA системе не гарантирует, что уже существующее программное обеспечение начнет использовать преимущества, предлагаемые новым сервисом. Зная эту проблему, ведущие IT компании предлагают использовать SOA вместе с EDA (Event-driven Architecture, Событийно-ориентированная архитектура). EDA – это другой архитектурный подход построения систем, в котором предполагается, что связи между компонентами должны быть построены на основе уведомлений о событиях. По существу, событие – это изменение состояние какого либо объекта, важного для системы. Связь между компонентами системы устанавливаются только в том случае, когда происходит уведомление о событии, произведенное, так называемым, генератором события. Эти события распространяются в промежуточной информационной среде и могут вызывать реакцию, так называемых, потребителей событий. По сравнению с SOA, генераторы и потребители событий полностью разъединены, а так же между ними возможен асинхронный обмен «многие ко многим». Промежуточная информационная среда отвечает за соответствующее распределение событий по системе.

Объединяя SOA и EDA, мы получаем в высокой степени привлекательную архитектуру для эффективного выполнения распределенных и полностью разделенных сервисов, которые могут эффективно выполняться как синхронно, так и асинхронно. Однако такая архитектура не может в полной мере обеспечить все современные потребности в построении распределенных систем. Информационная среда, основанная на событиях, является подходящим решением для интеграции разделенных систем, но на самом деле возникают большие трудности, связанные с тем, что крайне трудно создать такого рода среду в гетерогенной системе, а так же обеспечить унифицированность событий. Для проектирования новых систем необходимо ввести соглашения, согласно которому события, а именно их наличие, синтаксис и семантика, должны быть единообразными. Таким образом, адаптация, или, как еще модно сказать, унификация моделей данных остается одной из основных проблем.

Немаловажным остается вопрос обеспечения большей автоматизации процесса поиска веб-сервисов. Основой в этом направлении являются семантические web-технологии, которые позволяют работать с гетерогенными данными. Целью проект METEOR-S является расширение существующей технологий аннотирования, обнаружения, комбинирования и выполнения веб-сервисов на основе семантики. Главным направлением в этом проекте является объединение WSDL-S, совершенно нового подхода в аннотировании веб-сервисов на основе семантики, и использование методов планирования. OWL-S описывает онтологию для семантических веб-сервисов, основанных на языке OWL (Web Ontology Language).

Существует большое количество наработок в этом направлении, но все они являются разрозненными и не составляют единую систему. В конце хотелось бы отметить, что WSMX (Web Service Execution Environment) и IRS-III является проектами, основанными на WSMO (Web Service Modeling Ontology). Оба проекта WSMX и IRS-III направлены на реализацию семантической среды выполнения (SEE – Semantic Execution Environment).

Семантическая среда выполнения

Идея семантические аннотирование гетерогенных веб-сервисов вызвана желанием автоматизации процессов обнаружения, объединения и вызов веб-сервисов в сети интернет. Для реализации этих возможностей требуется соответствующая инфраструктуры программного обеспечения, которая может интерпретировать, и где возможно, выполнять семантические описания. Согласно всему вышесказанному требуется разработка инфраструктуры, которая будет поддерживать технологию семантических веб-сервисов и которая будет отвечать всем современным жестким требованиям разработки программного обеспечения. Решением именно этого вопроса было поручено заняться техническому комитету OASIS SEE. Их целью является создание базовой спецификации для семантической среды выполнения.

Основой семантической среды выполнения является объединение последних исследований в области семантических веб-сервисов с архитектурой поддержки сервис-ориентированных систем, которая более полно называется семантическая, основанная на событиях, сервис-ориентированная архитектура. Семантическая среда выполнения должна описывать основные архитектурные подходы при реализации поиска, объединения, посредничества и вызова веб-сервисов на основе их семантических аннотаций, используя концептуальную модель WSMO.

Фундаментальным принципом семантической среды выполнения является разделение ее компонентов. Компоненты не могут вызывать друг друга непосредственно через локальные или удаленные методы. Вместо этого в семантической среде выполнения используется событийный механизм, где компоненты подписываются на события. Компонента может являться потребителем события, т.е. каким-либо образом реагировать на него, или генератором, т.е. генерировать событие в систему. Таким образом, на сервисы, расположенные физически на разных серверах, не накладывается никаких ограничений, кроме того, что между ними должен существовать канал связи.

Аналогично семантической аннотации данных и сервисов, достижение полного семантического взаимодействия между компонентами семантической среды выполнения требует формализации событий. Мы предлагаем две категории событий: информационные и процедурные. Информационные события используются для обеспечения основных возможностей системы, а именно поиск сервисов, посреднические запросы и другие. Формализация информационных событий необходима для построения удобной среды взаимодействия. Процедурные события отвечают за состояние выполняемой процедуры, например, начало поиска. Целью такого рода событий может являться балансировка загрузки и отслеживания состояние системы.

Литература

<http://www.soapprinciples.com/>

<http://ibm.com>

<http://ieee.org>

ИССЛЕДОВАНИЕ ВЛИЯНИЯ СВОБОДНЫХ ПАРАМЕТРОВ ПАРАЛЛЕЛЬНОГО МЕТОДА ОПТИМИЗАЦИИ РОЕМ ЧАСТИЦ НА ЕГО ЭФФЕКТИВНОСТЬ

Селиверстов Е.Ю.

д.ф-м.н., профессор Карпенко А.П.

МГТУ им. Баумана, кафедра РК6, Москва, Российская Федерация

A STUDY OF INFLUENCE OF FREE PARAMETERS ON EFFICIENCY OF PARALLEL PARTICLE SWARM OPTIMIZATION METHOD

E.U. Seliverstov

Sc.D., professor A.P. Karpenko

Bauman Moscow State Technical University, RK6 department, Moscow, Russian
Federation

Аннотация

Предложен параллельный метод и алгоритм глобальной оптимизации роя частиц (PSO) с островной моделью параллелизма для графических SIMD-систем. Выработаны рекомендации по выбору свободных параметров алгоритма. Исследовано влияние свободных параметров на скорость сходимости, достигаемые экстремумы и масштабируемость по размерности задачи.

Abstract

Parallel method and algorithm of global particle swarm optimization (PSO) with island parallel model is proposed. Recommendations for choosing free algorithm parameters are given. A study of their impact to convergence speed, achieved extremums and problem dimension scalability is given.

1. Параллельный метод роя частиц

Метод роя частиц (PSO) - это новый метод эволюционной поведенческой глобальной многопараметрической оптимизации [1]. В работе предлагается метод GIPSO с островной многороевой моделью [2] для реализации на графических процессорах архитектуры CUDA.

Рассмотрим задачу глобальной безусловной минимизации функции $\Phi(X)$ в n -мерном арифметическом пространстве R^n : $\min \Phi(X) = \Phi(X^*)$, $X \in R^n$. Множество частиц обозначим $P = \{P_i, i \in [1: N]\}$, где N - размер популяции. Частица P_i определяется n -мерными векторами координат $X_{i,t}$ и скорости $V_{i,t}$. Итерации в методе PSO выполняются по схеме:

$$X_{i,t+1} = X_{i,t} + V_{i,t+1}; V_{i,t+1} = \chi(V_{i,t} + \phi \otimes (Y_{i,t}^g - X_{i,t})) \quad (1)$$

где

$$\chi = \frac{2k}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}, \phi = U[0, \phi_1] + U[0, \phi_2]; Y_{i,t}^g = \frac{U[0, \phi_1] \otimes X_{i,t}^b + U[0, \phi_2] \otimes X_{i,t}^g}{\phi} \quad (2)$$

Здесь $U[a, b]$ - n -мерный вектор псевдослучайных чисел, равномерно распределенных в интервале $[a, b]$; $X_{i,t}^b$ - вектор координат частицы P_i с наилучшим значением функции $\Phi(X)$ за все время поиска; $X_{g,t}$ - вектор координат соседней с данной частицы с наилучшим за время поиска значением $\Phi(X)$; χ, ϕ_1, ϕ_2 - свободные параметры алгоритма. Метод GIPSO использует разбиение роя на NS подроев S_k размером N_k . Каждые T_z итераций осуществляется миграция частиц в соответствии со стратегией репликации. Скорости частиц ограничиваются некоторыми предельными величинами $V^{max} = \gamma X^{max}$.

2. Свободные параметры параллельного метода роя частиц

Метод роя частиц отличается от многих других эволюционных методов оптимизации наличием большого количества свободных настраиваемых параметров. Такие параметры, как χ , γ , ϕ , NS , NN , N , T , стратегия миграции, существенно влияют на основные характеристики алгоритма - скорость сходимости, факт сходимости, исследованный объем пространства.

Некоторые параметры канонического метода, являющегося базовым для метода GIPSO, можно оценить по результатам экспериментального тестирования [4]. Специфичные для параллельного алгоритма свободные параметры N , NS , NN ограничиваются особенностями параллельной архитектуры и также влияют на скорость сходимости и достигаемые экстремумы [3].

Для графического процессора ограничения по памяти диктует размер разделяемой памяти $G_{shm} = 16384$ байт. Алгоритм использует $K_{shm} = 4(3nN + N + n + 1)$ байтов общей памяти на один рой. Необходимо выполнение ограничения $K_{shm} \leq G_{shm}$.

Общее количество регистров в одном мультипроцессоре графического процессора составляет $G_{reg} = 8192$. Пусть потоку в вычислительном ядре i требуется R_i регистров. Введем функцию $round(X, M)$, округляющую целое X до целого, кратного M . Тогда для вычислительного блока, состоящего из J потоков, количество требуемых регистров определяется как

$$R_{bi} = round\left(R_i \cdot round(J, 32), \frac{G_{reg}}{32}\right)$$

Отсюда вытекают следующие ограничения на свободные параметры алгоритма:

$$N \leq \frac{G_{shm} - n - 1}{3n + 1}; \quad R_{bi} \leq G_{reg} \quad i = 1 \dots 3$$

Размерность пространства n является определяющей в выборе параметров N и NS и при больших значениях практически ограничивает размер роя N . Количество роев NS можно выбирать независимо.

3. Влияние количества и размера роев на свойства алгоритма

Основные свойства глобального алгоритма оптимизации, по которым проводится анализ свободных параметров, - это достигаемый экстремум и скорость сходимости. Оптимизируемые функции выбираются из набора CEC-2009 [5].

Для оценки влияния числа роев NS на скорость сходимости эта величина пробегала ряд 2,4,8,16,32,64. Выявлено равномерное увеличение скорости сходимости по мере увеличения числа роев. Слабое изменение достигаемого экстремума вскоре после начала итераций связано с пропорциональным увеличением количества наилучших частиц во всех роях в процессе миграции посредством репликации.

Размер популяции для классического метода CPSO при малой размерности пространства поиска оказывает слабое влияние на сходимость и выбирается по рекомендациям работы [4]. В методе GIPSO эта связь неоднозначна. Наше исследование показывает, что как и для метода CPSO, при малой размерности пространства ($n = 4$) размер роя слабо влияет на скорость сходимости. Приемлемые по скорости результаты показывают рои уже из 4...8 частиц. На всех тестовых функциях метод сходится. Для пространства большей размерности ($n = 8$) размер роя начинает оказывать влияние не только на скорость сходимости, но и на величину достигаемого экстремума. Минимальный размер роя, способного найти глобальный минимум всех тестовых функций, составляет 8 частиц.

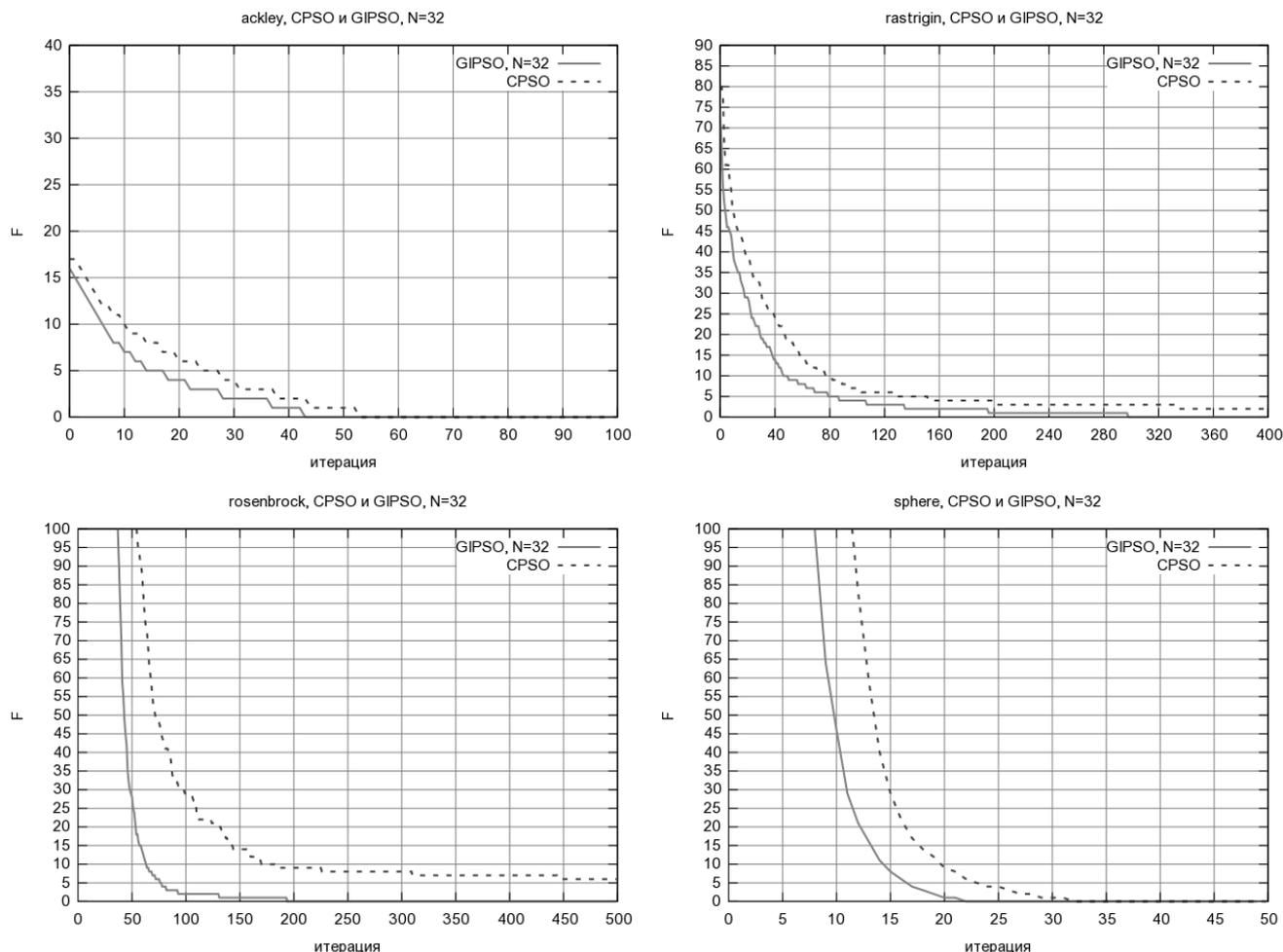


Рис. 1. Влияние размера популяции на скорость сходимости

4. Сравнительная оценка сходимости для алгоритмов GIPSO и CPSO

Сравнительное исследование сходимости указанных алгоритмов представлено экспериментом, в ходе которого определяется количество итераций I , необходимое для достижения заранее известной величины глобального минимума $g_f = 0.1$. Размерности пространства $n = 4, 8, 32$. На рис. 1 показаны результаты эксперимента для параметров $N = 16$, $NS = 4$, $n = 8$.

Эксперименты показывают, что алгоритм CPSO для простых задач позволяет гарантированно найти минимум уже с 16 частицами в рое. Однако на сложных задачах ($n = 32$) даже 64 частиц недостаточно для нахождения минимума по крайней мере одним тестом. Алгоритм GIPSO испытывает те же проблемы при малом числе роев. Увеличение числа роев при сохранении их размера постепенно уменьшает I . При большой размерности пространства алгоритм сходится не для всех задач.

Сравнение алгоритмов при одинаковых параметрах n и N , но при различных количествах подроев, показывает преимущество алгоритма GIPSO.

5. Масштабируемость алгоритма GIPSO по данным

Под масштабируемостью алгоритма по данным понимают способность алгоритма эффективно решать задачу при увеличении ее размера. Выполнено исследование поведения алгоритма при повышении количества роев. Масштабируемость при этом оценивается зависимостью относительного времени работы алгоритма P от количества роев NS , где $P = t/NS$. Параметр NS варьировался в широких пределах. Использовался фиксированный размер роя: $N=16$. На рис. 2 приведены некоторые результаты исследования масштабируемости алгоритма для известной функции Розенброка.

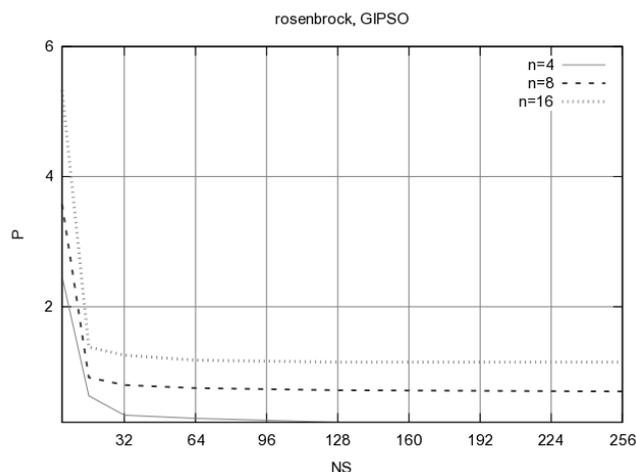


Рис. 2. Масштабируемость алгоритма GIPSO

Для алгоритма GIPSO возможно увеличение количества роев со строго линейным ростом скорости обработки. Это позволяет при переходе к более сложным задачи оптимизации свободно увеличивать количество роев. Конечно, следует провести анализ поведения большого количества роев при различных стратегиях миграции и не допустить преждевременного схождения процесса поиска.

Литература

1. *Kennedy, J.* Particle swarm optimization / J. Kennedy, R. Eberhart // In Proceedings of IEEE International conference on Neural Networks. — 1995. — Pp. 1942–1948.
2. *Belal, M.* Parallel models for particle swarm optimizers / M. Belal, T. El-Ghazawi // IJICIS. — 2004. — January. — V. 1. — Pp. 100–111.
3. NVIDIA Corporation. — CUDA Programming Guide Version 2.1, 2008 [Электронный ресурс]. URL: http://developer.nvidia.com/object/cuda_2_1_downloads.html (дата обращения 20.02.2010).
4. *Li-ping, Z.* Optimal choice of parameters for particle swarm optimization / Z. Li-ping, Y. Huan-jun, H. Shang-xu // Journal of Zhejiang University Science. — 2005. — Part 06. — Vol 6, no.6. — Pp. 528–534.
5. Benchmark functions for the CEC'2008 special session and competition on large scale global optimization: Tech. rep. / K. Tang, X. Yao, P. N. Suganthan et al.: Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.

SIMULINK-МОДЕЛИРОВАНИЕ МНОГОСЕКЦИОННОГО МАНИПУЛЯТОРА ТИПА ХОБОТ

Антонов Д.А.

Научный руководитель: д.ф.-м.н., профессор Карпенко А.П.

МГТУ им. Н.Э. Баумана, кафедра РКБ, г. Москва, Россия

SIMULATION OF MULTISECTION ARM TYPE PARALLEL ROBOT USING MATLAB AND SIMULINK

Antonov D.

Professor A.P. Karpenko

Bauman Moscow State Technical University, Moscow, Russia

Аннотация

Работа посвящена созданию моделей механизмов типа хобот на базе параллельных манипуляторов с шестью степенями свободы.

Abstract

The paper is dedicated to creation of arm type mechanisms models using parallel manipulators with six degrees of freedom.

В настоящее время имеется значительное количество областей науки, техники, медицины и пр., в которых возможностей традиционных роботов-манипуляторов недостаточно. Примерами таких областей может служить контроль качества трубопроводов изнутри труб, а также механическая обработка изделий сложной формы (например, штампов, пресс-форм, лопаток турбин и т.д.), при которой требуется перемещение инструмента по пяти-шести координатам.

Известно значительное количество проектов, посвященных разработке нетрадиционных манипуляционных роботов. Эти роботы используют аналоги одних из самых интересных «изобретений» природы - хоботов позвоночных и щупалец беспозвоночных животных.

Обычно механизм с параллельными кинематическими связями (в дальнейшем – параллельный манипулятор) состоит из подвижной платформы, которая прикрепляется к неподвижному основанию несколькими штангами, длина которых может управляется некоторым приводом. Вследствие того, что внешняя нагрузка на подвижную платформу разделяются между штангами, параллельные манипуляторы имеют хорошую грузоподъемность и при этом обеспечивать высокую точность.

Исследуемый механизм представляет собой семь последовательно соединенных секций, конструктивно представляющих собой платформы Стюарта-Гофа. Основание первой ступени является неподвижным. Каждая из последующих секций имеет в качестве своего основания подвижную платформу предыдущей секции. Все тела считаются твердыми.

Размеры первой ступени манипулятора взяты, исходя из параметров реально существующего промышленного робота FANUC F-200iB[4].

На рисунке 1 представлена схема системы управления семисекционным манипулятором на основе платформ Стюарта-Гофа (гексапода) [3]. Здесь блоки **1.1, ... n.1, ... 7.1** - пропорционально – интегрально - дифференциальные регуляторы соответствующей секции манипулятора [1]; **1.2, ... n.2, ... 7.2** – блоки задания траектории секций (эти блоки служат для получения по заданной траектории координат шарниров на подвижной платформе); **1.3, ... n.3, ... 7.3** – блоки моделей механизма секции, а **4** – блок построения графиков перемещения, скорости и ускорения седьмой платформы. Блок “Земля” реализует «землю», то есть инерциальную систему координат.

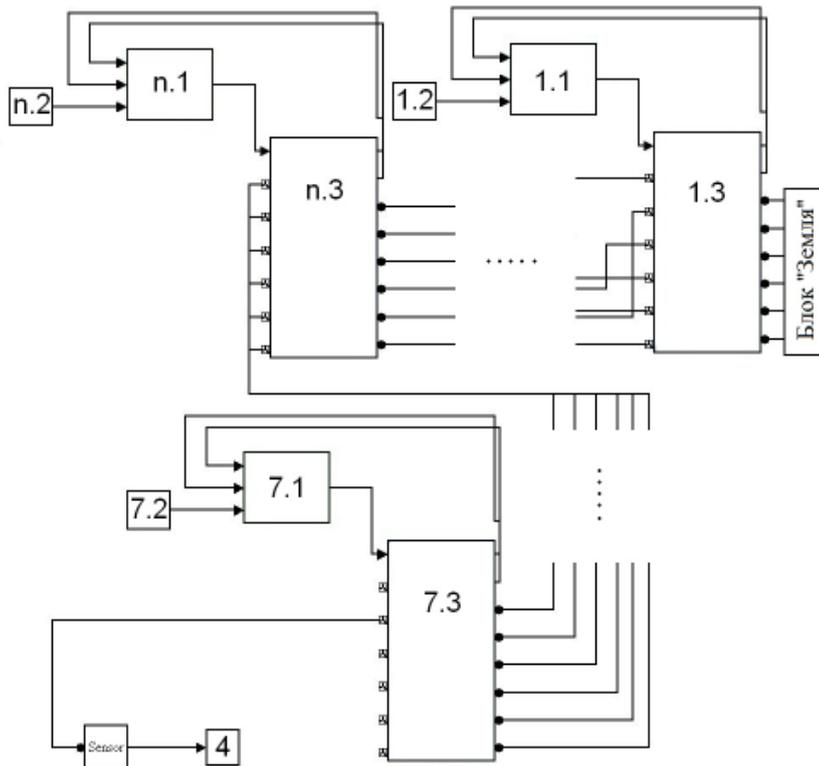


Рис.1. Схема системы управления семисекционным манипулятором на основе гексапода.

Ставится задача исследовать кинематику и динамику указанного семисекционного манипулятора [2]. В качестве входного воздействия рассматривается ступенчатое увеличение длины одной из штанг первой секции механизма.

На рисунке 2 изображена общая схема механизма “хобота” до подачи управляющего сигнала и после. Соответствующие изменения координат одного из шарниров подвижной платформы седьмой секции манипулятора иллюстрирует рисунок 3. Характер движения точки A_8 отвечает ожидаемому движению, обусловленному поворотом твердого тела вокруг некоторой оси: наблюдаем значительное перемещение указанной точки вдоль оси X , сопровождающееся увеличением, а затем уменьшением значения координаты Z .

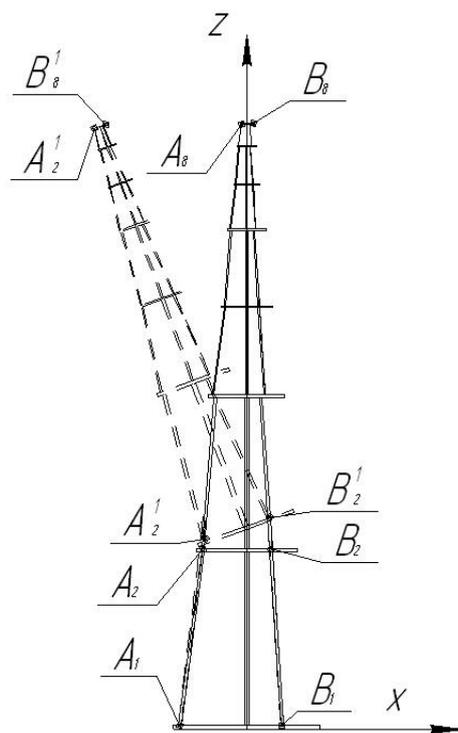


Рис.2. Положение механизма в плоскости XZ до и после подачи управляющего сигнала.

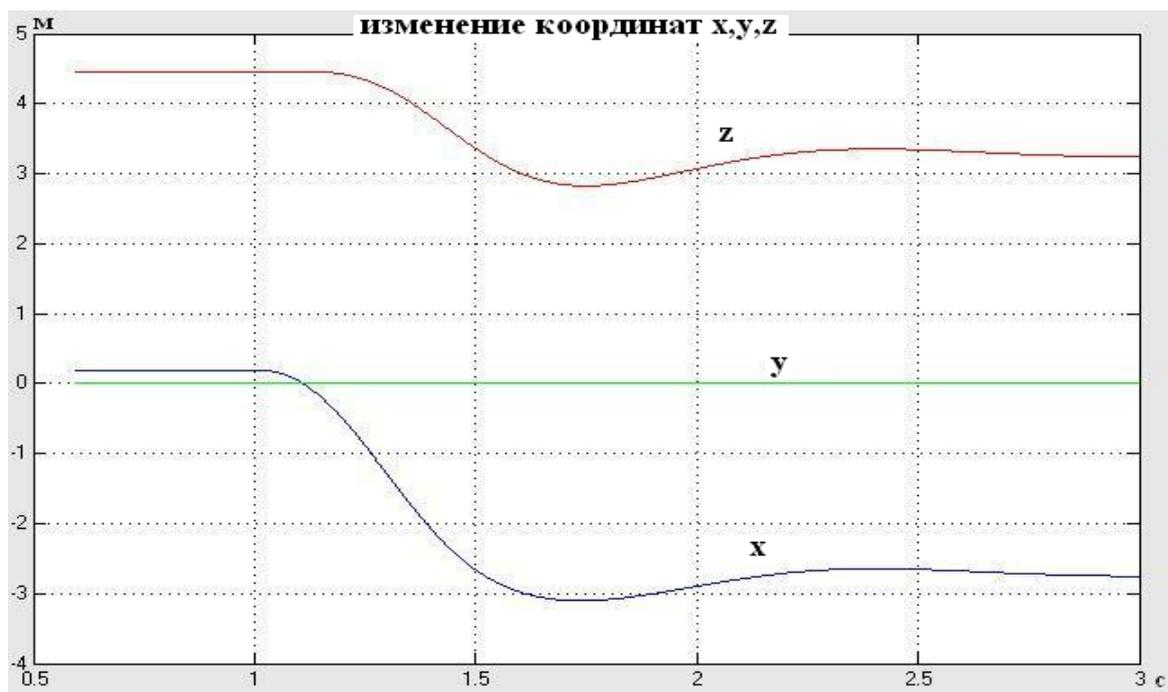


Рис.3. График изменения координат точки A_8 .

Некоторые результаты исследования представлены на рисунке 4. График отображает изменение силы реакции опоры (шарнира A_1 , расположенного на основании первой секции манипулятора) после подачи ступенчатого управляющего сигнала. Время калькуляции 3 секунд модельного времени составило порядка 3 минут, что можно считать приемлемым результатом.

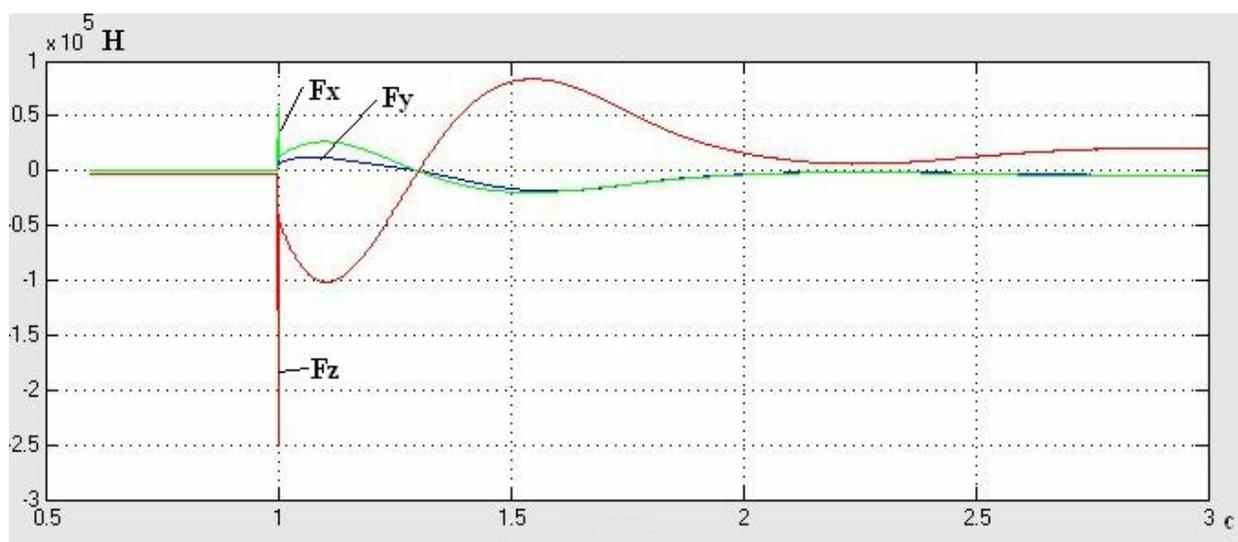


Рис.4. Силы в шарнирах A₁ первой секции манипулятора на основе гексапода.

Несмотря на возникающие осцилляции сил реакции опор, переходный процесс сходится за реальное время, что позволяет говорить об обоснованности использования ПИД-регуляторов. Качественно, полученная картина также соответствует представлению о реальном переходном процессе для механизма с абсолютно твердыми телами и упругими связями.

Литература

1. Дьяконов, В.П. MATLAB 6.5 SP1/7.0 Simulink 5/6 в математике и моделировании. /В.П. Дьяконов – М.: СОЛОН-Пресс, 2005. – 405с.
2. J.P.Merlet. Parallel Robots. Solid mechanics and its applications.- Kluwer Academic Publishers, V. 74, 2000.
3. Махов А.А. Моделирование механических систем с помощью пакета расширения SimMechanics [Электронный ресурс] / (<http://exponenta.ru/educat/systemat/mahov/simmechanics.asp>).
4. The Robot Company [Электронный ресурс] / (<http://www.robotco.com/fanuc-f200.htm>).

УЧЕТ МОРФОЛОГИИ ЯЗЫКА ПРИ ПРОЕКТИРОВАНИИ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННО-ПОИСКОВЫХ СИСТЕМ

Татьян Т.С.

Научный руководитель: к.т.н., доцент Волосатова Т.М.
МГТУ им. Н.Э. Баумана, Кафедра «САПР» (РК6), Москва, Россия.

MORPHOLOGY HANDLING IN INTELLIGENT INFORMATION RETRIEVAL SYSTEMS DESIGN

Tatyan T.S.

Scientific chief: PhD, associate professor Volosatova T.M.
CAD/CAE Department (RK6) of BMSTU, Moscow, Russia.

Аннотация

Одним из основных критериев качества работы современной информационно-поисковой системы является поддержка морфологии языка. Распознавание морфологии позволяет пользователю делать запросы на естественном языке, что для человека является наиболее привычным и удобным. В данной работе рассматриваются основные особенности морфологии русского и английского языков и подходы к их анализу и учету в современных интеллектуальных ИПС.

Abstract

Main features of Russian and English morphology are listed and contrasted. Approaches to morphology handling in state-of-the-art intelligent information retrieval systems are given.

Введение

В повседневной жизни большинство из нас постоянно сталкивается с задачами информационного поиска. Значительную роль в решении данного вида задач играют различные поисковые системы, в частности, системы, осуществляющие поиск информации в сети Интернет, в которой в настоящее время улучшение качества информационного поиска является одним из наиболее приоритетных направлений развития.

Одним из основных критериев качества работы современной информационно-поисковой системы (ИПС) является поддержка морфологии языка, используемого для взаимодействия с подобной системой. Распознавание морфологии естественного языка, например, русского или английского, позволяет пользователю делать запросы к системе на естественном языке, что для человека является наиболее привычным и удобным.

По статистике, для общения с поисковой системой чаще всего используются существительные – так, типичный поисковый запрос в русскоязычном Интернете состоит из 2-3 существительных, написанных кириллицей. Следовательно, первоочередная задача модуля распознавания морфологии языка - возможность работы с существительными. Однако, морфология и словообразование сильно зависят от языка, на котором производится запрос. В связи с этим, подходы к распознаванию и анализу морфологии различных языков должны отличаться. При этом общий алгоритм работы ИПС выглядит следующим образом:

1. Удаление стоп-слов (и, или, но...), которые заданы списком
2. Приведение слов, составляющих документ, к нормальной морфологической форме
3. Построение вектора терминов документа
4. Сравнение вектора запроса с векторами документов
5. Выдача документов, функция подобия которых превышает некоторый порог

Морфология имен существительных русского и английского языков

По способу образования словоформ русский язык - флективный, для него характерна развитая система словоизменения с помощью различного рода флексий (от лат. flexio - сгибание), или окончаний. Русский язык относится также к классу синтетических языков: в слове объединено и лексическое, и грамматическое значение. Имя существительное в русском языке обладает категориями рода (выделяют мужской, женский, средний, общий

род), одушевлённости, числа (различают единственное и множественное число) и падежа (6 основных падежей, существительное может принимать шесть падежей в единственном числе, и шесть – во множественном). Таким образом, одно и то же имя существительное может встречаться в 12 различных формах в одном и том же документе. Кроме того, некоторые словоформы могут образовываться с помощью изменения основы слова, например “что – ложь; чего – лжи”. Для существительного в русском языке леммой (нормальной, словарной формой) является словоформа, соответствующая именительному падежу (единственного числа, если слово изменяется по числам; если оно изменяется по родам - мужского рода).

В отличие от русского языка, в английском языке практически отсутствует словоизменение в зависимости от роли слова, в том числе по падежам; падежные отношения передаются позицией слов в предложении и предложными конструкциями, при этом основа слова изменяется только в исключительных случаях, например, в неправильных глаголах.

Псевдоморфология в примитивных системах

Модуль учета морфологии языка может являться частью различных программных комплексов, работающих с естественным языком – начиная от различных простых текстовых процессоров и заканчивая сложными ИПС. Одним из подходов к проектированию модуля анализа морфологии языка в несложных системах можно считать использование псевдоморфологии, что часто позволяет достичь приемлемого результата без необходимости приведения слова к нормальной морфологической форме, т.е. выделения основы или леммы заданного слова. Примером такой системы может служить прикладная программа, подсчитывающей плотность указанного слова в документе с учетом всех его возможных словоформ. В этом случае оптимальным решением может стать применение стандартных для выбранного языка программирования функций сравнения строк последовательно к заданному слову и ко всем словам документа. Так, для языка PHP, получившего широкое распространение в среде веб-разработок, такими функциями могут являться функция `similar_text`, использующая алгоритм Оливера, или функция, рассчитывающая расстояние Левенштейна. Главным достоинством данного подхода является его простота, главным же его недостатком - погрешности полученных результатов, причем часто достаточно существенные. Применение более сложных подходов, позволяет получать более точные и корректные результаты, а, следовательно, на порядок увеличивает эффективность работы модуля учета морфологии языка.

Анализ морфологии английского и русского языков в сложных ИПС

Для анализа английских текстов поисковые машины используют различные вариации стемминга. Стемминг (англ. *stemming*) - это процесс нахождения основы слова для заданного исходного слова, при этом основа слова необязательно совпадает с его морфологическим корнем. В большинстве случаев, стемминг сводится к удалению флексий, в некоторых алгоритмах для нахождения основы слова используются таблицы соответствий между основой слова и допустимыми вариантами флексий. Однако из-за особенностей, присущих русскому языку, для эффективного анализа русскоязычного текста поисковые машины должны использовать более сложные лемматизирующие алгоритмы, в которых, в отличие от алгоритмов стемминга, выделение основы слова происходит в 2 этапа, а правила его получения зависят от части речи, к которой принадлежит слово. Так, первоначальным этапом процесса лемматизации является определение части речи анализируемого слова, к которому затем применяются правила нормализации, характерные именно для данной части речи, в результате чего получается лемма заданного слова.

В русском языке получение леммы слова, как правило, является сложной задачей из-за обилия правил словоизменения и словообразования и наличия исключений. В настоящее время при разработке модулей учета морфологии русского языка в основном применяется

разработанная А.А. Зализняком система классификации слов русского языка, т.к. данная система позволяет алгоритмизировать как процесс получения исходной формы слова из словоформ, так и обратный процесс генерации допустимых словоформ из исходной формы слова. Подход к проведению морфологического анализа на основе словаря Зализняка часто называют «четкой» морфологией.

Основные сведения о словаре Зализняка

Грамматический словарь русского языка (словарь Зализняка) — составленный А. А. Зализняком словарь примерно 110 тыс. базовых словоформ русского языка с их полным морфологическим описанием. Это основополагающий труд по морфологии, где впервые был предложен системный подход к описанию грамматических парадигм, включающих не только изменение буквенного состава слов, но и ударения. Словарь впервые был издан в 1977 г., с тех пор неоднократно переиздавался.

Благодаря предложенному Зализняком системному подходу стала возможной алгоритмизация процесса словоизменения слов русского языка, а, следовательно, программирование автоматизированных систем, взаимодействующих с русской морфологией. Электронная версия этого словаря легла в основу большинства современных компьютерных программ, работающих с русской морфологией (системы проверки орфографии, автоматического перевода, реферирования и т. п.).

В словаре принят не обычный алфавитный порядок расположения слов, а так называемый инверсионный. Расположение слов в обычном алфавитном порядке означает, что из произвольных двух слов в словаре раньше встретится то, у которого начальная буква ближе к началу алфавита; при одинаковых начальных буквах сравниваются вторые от начала буквы, если и они одинаковы, то третьи и т. д. При инверсионном алфавитном порядке используется тот же принцип, с тем единственным отличием, что сравнение двух слов начинается не с начальных букв, а с конечных, и далее продолжается от конца слова к началу. Такой порядок расположения слов в словаре Зализняка наиболее целесообразен и нагляден, т.к. обращает внимание читателя на тот факт, что в русском языке слова, оканчивающиеся одинаковой комбинацией букв, как правило, изменяются (склоняются/спрягаются) одинаково, т.е. инверсионный порядок позволяет увидеть систему в многочисленных правилах словоизменения.

Каждое слово представлено в словаре своей исходной, или словарной, формой (которая образует так называемое заглавное слово статьи). Для склоняемых частей речи это именительный падеж (если слово изменяется по числам — единственного числа; если оно изменяется по родам - мужского рода), для глаголов — инфинитив, для неизменяемых частей речи — единственная имеющаяся форма.

Словарная статья в общем случае состоит (не считая заглавного слова) из: 1) основного буквенного символа, 2) индекса, 3) дополнительных помет и указаний (в частных случаях тот или иной из этих элементов может отсутствовать) — т.е. для каждого слова указан определенный код. Кроме того, известна система правил, с помощью которой можно построить все формы данного слова, отталкиваясь от начальной словоформы и соответствующего ей кода. Помимо построения каждой словоформы, система правил автоматически ставит в соответствие ей морфологические характеристики.

Морфологический анализ на основе словаря Зализняка

При проведении четкого морфологического анализа необходимо иметь словарь всех слов и всех словоформ языка. Данный словарь можно построить на основе словаря Зализняка по очевидному алгоритму: перебрать все слова из словаря, для каждого из них определить все возможные их словоформы и занести их в формирующийся словарь.

При таком подходе для проведения морфологического анализа заданного слова (рис. 1) необходимо просто найти его в словаре, где уже хранятся точные, «окончательно известные» значения всех его морфологических характеристик.



Рис. 1. Морфологический анализ на основе словаря Зализняка

Как уже было упомянуто, словарь А. А. Зализняка включает около 110 000 слов. Однако в русском языке их больше, кроме того, постоянно появляются новые. Эти слова следует обрабатывать, используя описанные в словаре Зализняка словообразовательные типы. Результат обработки будет тем лучше, чем больше новое слово похоже на обычные слова языка.

Выводы

Морфология и словообразование сильно зависят от естественного языка, который используется при работе с ИПС. В связи с этим, подходы к распознаванию и анализу морфологии различных языков должны отличаться. Кроме того, подходы к реализации модуля учета морфологии языка должны выбираться исходя из сложности проектируемой системы, а также поставленных перед ней целей и задач. Так, в несложных системах целесообразно использовать псевдоморфологические подходы, при работе с русским языком – систему на основе словаря Зализняка и лемматизирующие алгоритмы.

В силу специфики написания запросов к ИПС, основной задачей модуля анализа морфологии языка в ИПС должна являться работа с именами существительными.

Литература

1. Зализняк А.А. Грамматический словарь русского языка: Словоизменение. – Изд. 6-е, стер. – М.: АСТ-ПРЕСС КНИГА, 2009. 800 с. – (Фундаментальные словари).
2. PHP Manual / Edited by G. Hojtsy. The PHP Documentation Group, 2006.
3. <http://company.yandex.ru/facts/researches>
4. <http://en.wikipedia.org/wiki/Lemmatization>
5. <http://en.wikipedia.org/wiki/Stemming>
6. http://ru.wikipedia.org/wiki/Английский_язык
7. http://ru.wikipedia.org/wiki/Русский_язык
8. <http://www.osp.ru/os/2003/12/183694/>
9. <http://www.searchengines.ru/articles/005047.html>
10. <http://www.seo-copywrite.ru/14/>

ВЫСОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛЕНИЯ ДЛЯ НАНОИНЖЕНЕРИИ

Тимофеев Г. Г.

Научный руководитель: *Власов А. И.*
МГТУ им. Н.Э. Баумана, кафедра ИУ4, Россия, Москва.

HIGH-PERFORMANCE CALCULATIONS FOR NANOENGINEERING

Timofeev G. G.

Scientific Leader: *Vlasov A. I.*
Russia, Moscow, Bauman MSTU, IU4

Аннотация

В данной работе рассмотрены различные виды кластеров, использование высокопроизводительных вычислений для моделирования задач нанотехнологии, структура кластера НОЦ «Наносистемы», так же приведена инструкция работы на нем.

Abstract

In this scientific work has been considered different kinds of clusters, using of High-performance calculations for modeling of problems of nanoengineering, structure of cluster of NOS “Nanosystems”, also has been considered manual for working on it.

Введение

Кластер — группа вычислительных модулей, объединённых высокоскоростными каналами связи и представляющая с точки зрения пользователя единый аппаратный ресурс.

Целью работы является построение единой вычислительной системы НОЦ, которая позволяет решать текущие задачи нанотехнологии.

Ранее, для расчетов, применялись стационарные компьютеры, но их производительность была достаточно мала, из-за чего затрачивалось большое время на работу с математическими задачами. Кроме того, из-за дискретности такой системы, возникают проблемы с единством данных, т.е. одна часть задачи решалась на одном компьютере, другая – на другом, что приводит к затруднениям – из-за различных ОС и ПК может возникнуть несовместимость частей, не говоря уже о неудобстве переноса данных и последующему обращению к ним. В единой же вычислительной системе, когда реализована система параллельной работы, несколько пользователей работают на одном вычислительном узле, и данные соответственно хранятся на нем же, проблема с несоответствием версий решена по умолчанию. Так же удобно обращение к этим данным, так как способ подключения к узлу унифицирован и достаточно прост. В любой момент можно посмотреть состояние задачи, которую решает кластер, загруженность вычислительной системы на данный момент, что позволяет оценить примерные сроки выполнения задания.

1. Классификация кластеров.

Рассмотрим классификацию кластеров по принципу действия и применению (Рис. 1) [1].



Рис. 1. Классификация кластеров.

Кластеры высокой доступности.

Обозначаются аббревиатурой НА (англ. *High Availability* — высокая доступность). Создаются для обеспечения высокой доступности сервиса, предоставляемого кластером. Избыточное число узлов, входящих в кластер, гарантирует предоставление сервиса в случае отказа одного или нескольких серверов. Типичное число узлов — два, это минимальное количество, приводящее к повышению доступности. Создано множество программных решений для построения такого рода кластеров. В частности, для OpenVMS, GNU/Linux, FreeBSD и Solaris существует проект бесплатного ПО Linux-NA.

Кластеры распределения нагрузки.

Принцип их действия строится на распределении запросов через один или несколько входных узлов, которые перенаправляют их на обработку в остальные, вычислительные узлы. Первоначальная цель такого кластера — производительность, однако, в них часто используются также и методы, повышающие надёжность. Подобные конструкции называются серверными фермами. Программное обеспечение (ПО) может быть как коммерческим (OpenVMS, MOSIX, Cluster, Platform LSF HPC, Sun Grid Engine, Moab Cluster Suite, Maui Cluster Scheduler), так и бесплатным (Linux Virtual Server).

Вычислительные кластеры.

Кластеры используются в вычислительных целях, в частности в научных исследованиях. Для вычислительных кластеров существенными показателями являются высокая

производительность процессора в операциях над числами с плавающей точкой (flops) и низкая латентность объединяющей сети, и менее существенными — скорость операций ввода-вывода, которая в большей степени важна для баз данных и web-сервисов. Вычислительные кластеры позволяют уменьшить время расчетов, по сравнению с одиночным компьютером, разбивая задание на параллельно выполняющиеся ветки, которые обмениваются данными по связывающей сети. Одна из типичных конфигураций — набор компьютеров, собранных из общедоступных компонентов, с установленной на них операционной системой Linux, и связанных сетью Ethernet, Myrinet, InfiniBand или другими относительно недорогими сетями. Такую систему принято называть кластером Beowulf. Специально выделяют высокопроизводительные кластеры (Обозначаются англ. аббревиатурой HPC Cluster — High-performance computing cluster). Список самых мощных высокопроизводительных компьютеров (также может обозначаться англ. аббревиатурой HPC) можно найти в мировом рейтинге TOP500. В России ведется рейтинг самых мощных компьютеров СНГ TOP50 Суперкомпьютеры.

Системы распределенных вычислений.

Такие системы не принято считать кластерами, но их принципы в значительной степени сходны с кластерной технологией. Их также называют grid-системами. Главное отличие — низкая доступность каждого узла, то есть невозможность гарантировать его работу в заданный момент времени (узлы подключаются и отключаются в процессе работы), поэтому задача должна быть разбита на ряд независимых друг от друга процессов. Такая система, в отличие от кластеров, не похожа на единый компьютер, а служит упрощенным средством распределения вычислений. Нестабильность конфигурации, в таком случае, компенсируется большим числом узлов.

2. Функциональная схема кластера НОЦ.

Кластер НОЦ содержит узлы, которые решают различные задачи (Рис. 2).

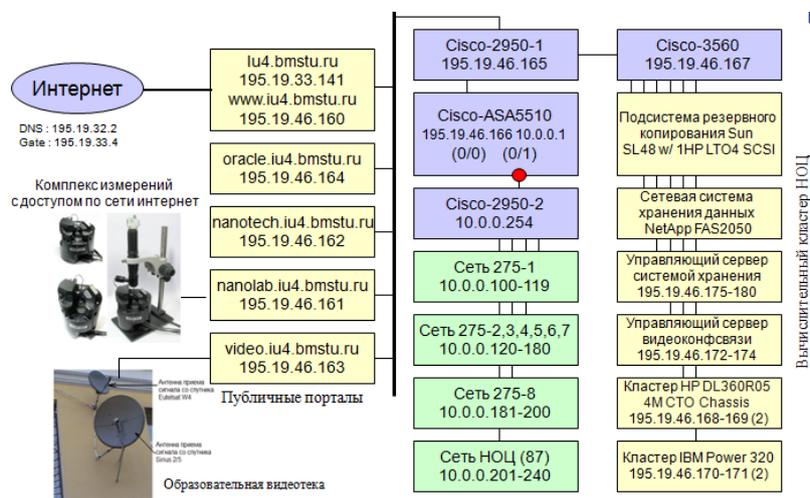


Рис. 2. Функциональная схема кластера НОЦ.

На схеме приведен общий вид информационной системы НОЦ. Iu4.bmstu.ru – узел с порталом, который содержит информацию о проектах кафедры, информацию для абитуриентов, контакты для связи;

Oracle.iu4.bmstu.ru – портал, посвященный конструкторско-технологическому проектированию на основе СУБД Oracle;

Nanotech.iu4.bmstu.ru – портал лаборатории наноинженерии НОЦ «Наносистемы»;

Nanolab.iu4.bmstu.ru – подсистемы для удаленной работы с экспериментальным оборудованием;

Video.iu4.bmstu.ru – подсистема, предназначенная для обучения путем просмотра образовательных фильмов, посвященных технологии производства электронных средств;

Вычислительный кластер НОЦ – подсистема для решения математических вопросов ТПЭС.

3. Общесистемное ПО вычислительной подсистемы.

Вычислительная подсистема стоит на основе Linux и программного пакета Oscar (Рис. 3).

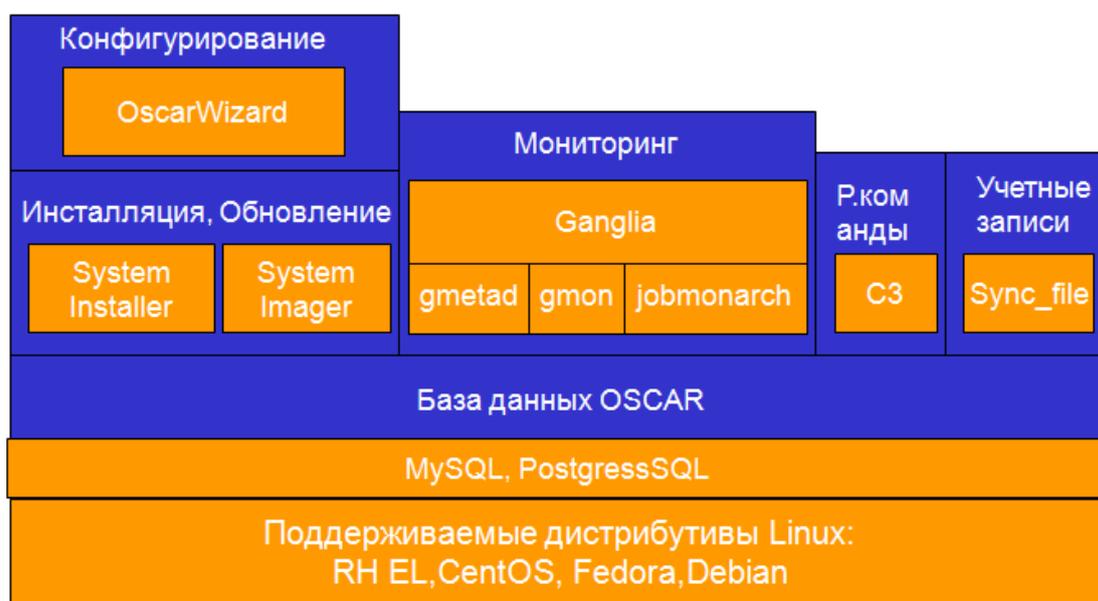


Рис. 3. Общесистемное ПО вычислительного кластера.

Модульная структура общесистемной организации вычислительной подсистемы. Основой является выбор дистрибутива Linux. Для построения кластера на основе программного пакета Oscar рекомендуются Red Hat-подобные дистрибутивы: Red Hat Enterprise Linux, CentOS, Fedora. Для хранения информации о текущих пользователях и их задачах, Oscar использует СУБД, предпочтительно использовать MySQL вследствие доступности и свободы последнего. Для построения кластера и последующего его реконфигурирования используется модуль OscarWizard. За мониторинг отвечает модуль Ganglia, который позволяет в реальном времени получить информацию об «очереди» на выполнение задачи, степень загруженности узлов, а так же дает возможность снять задачу из «очереди».

4. Подключение к кластеру.

Для подключения подходит как Linux – система, так компьютер с ОС Windows (Рис. 4).

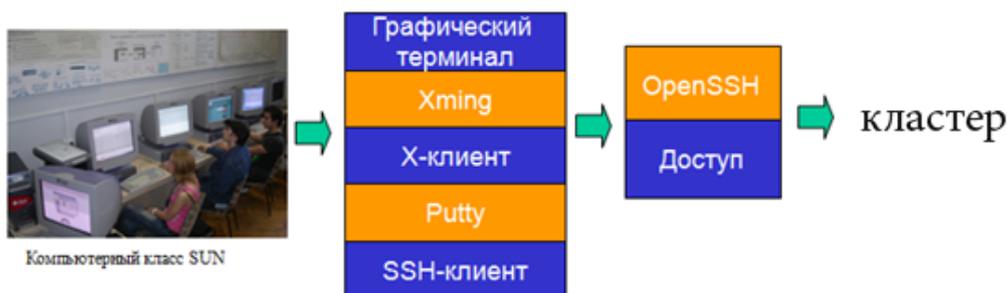
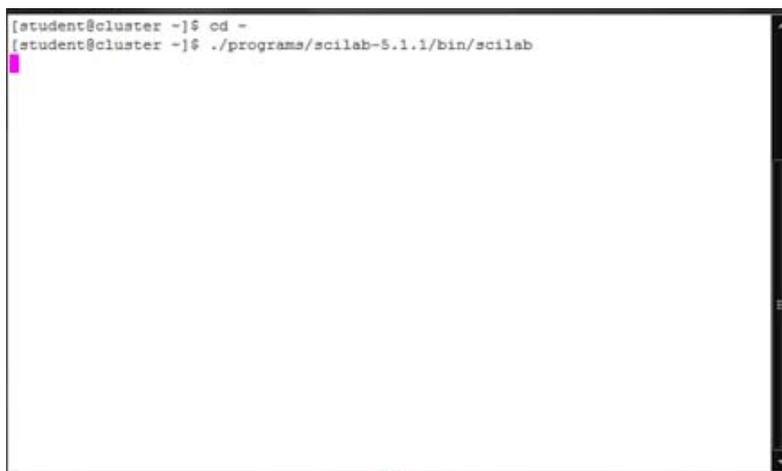


Рис. 4. Подключение к кластеру.

Используя SSH-клиент можно получить доступ к вычислительным ресурсам системы с любого компьютера с Интернетом. После подключения, находясь в домашнем каталоге пользователя, за которого совершен вход, можно приступить к выполнению задач (Рис. 5).



```
[student@cluster ~]$ cd -
[student@cluster ~]$ ./programs/scilab-5.1.1/bin/scilab
```

Рис. 5. Выполнения команды на запуск ПП Scilab из домашнего каталога пользователя «student».

Например, перейдя в каталог с ПП Scilab и прописав команду “./scilab” запустится диалоговое окно этого программного пакета (Рис. 6).

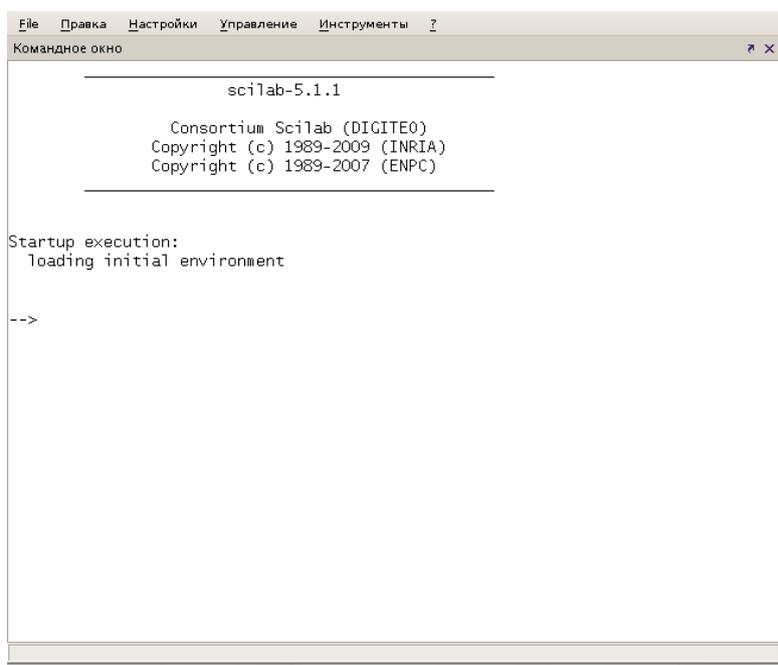


Рис. 6. ПП Scilab.

Так же на кластере возможно моделирование в химико-математическом пакете NWChem, расчет запрещенных зон фотонных кристаллов в пакете MPB и в других пакетах моделирования.

Литература

1. Gregory F. Pfister In Search of Clusters

ОПТИМАЛЬНАЯ ГЕОМЕТРИЯ МНОГОСЕКЦИОННОГО МАНИПУЛЯТОРА ТИПА «ХОБОТ»^{*)}

Волкоморов С.В.

д.ф.-м.н., проф., Карпенко А.П.

МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Россия.

OPTIMAL GEOMETRY OF MULTISECTION «TRUNK» MANIPULATOR

S.V. Volkomorov

D. Sc., prof. A.P. Karpenko

MSTU n.a. Bauman, Moscow

Аннотация

Сформулированы задачи проектирования формы многосекционного манипулятора типа «хобот». Рассмотрен подход к решению одной из них. Выведено интегральное уравнение для нахождения зависимости радиуса манипулятора от координаты сечения. Произведено численное решение поставленной задачи.

Abstract

Design tasks for multisection «trunk» manipulator shape are posed. Solving approach for one of them is considered. Integral equation for manipulator radius coordinate-dependent function is obtained. Numerical solution is performed.

В настоящее время в машиностроении актуальной является задача разработки технологических машин для выполнения механической обработки внутренних поверхностей полостей сложной формы, например, внутренних каналов охлаждаемых лопаток турбин для авиационной и космической техники. Обычно для решения перечисленных задач используются многосекционные рычажные манипуляторы либо гибкие манипуляторы.

Серьезным недостатком указанных манипуляторов является их недостаточная жесткость, усложняющая управление ими, затрудняющая использование высокоэнергетического обрабатывающего инструмента и достижение высокой точности обработки.

В значительной мере преодолеть указанные недостатки могут манипуляторы типа "хобот", построенные на основе многосекционных механизмов с параллельной структурой [1]. Варианты структуры секций манипулятора типа «хобот» рассмотрены в работе [2].

Одними из первых при конструировании многосекционного манипулятора типа «хобот» возникают следующие задачи: 1) найти оптимальную форму секций манипулятора; 2) найти их оптимальные размеры.

При решении первой из указанных задач в работе используется подход на основе представления манипулятора в виде консольной балки круглого поперечного сечения, плотность материала которой постоянна и равна c .

Рассмотрим консольную балку, которая функционирует в условиях изгиба и в которой имеют место только нормальные напряжения. Максимальное напряжение в сечении X

балки при изгибе в плоскости Oxy в этом случае равно $y_{\max}(X) = \frac{M_o(X)}{W_z(X)} = a_1 \leq [y]$,

^{*)} Работа выполнена в рамках аналитической ведомственной целевой программы «Развитие потенциала высшей школы (2009 – 2010 годы)», проект 2.1.2/1509.

где $M_o(X)$ - изгибающий момент в сечении X , $W_z(Z)$ - момент сопротивления данного сечения, $[y]$ - максимально допустимое напряжение (рис. 1). Отсюда следует, что у балки равного сопротивления изгибу момент сопротивления сечения должен быть прямо пропорционален изгибающему моменту в этом сечении и для сплошной балки круглого сечения можно записать

$$W_z(X) = \frac{M(X)}{a_1} = \frac{pr^3(X)}{4},$$

$$r(X) = 3\sqrt{\frac{4M_o(X)}{pa_1}},$$
(1)

где $r(X)$ - радиус балки в сечении X .

Последовательно рассмотрим нагружение балки внешним сосредоточенным моментом, внешней сосредоточенной силой, распределенной силой веса балки.

1. Нагружение сосредоточенным моментом

Из выражения (1) следует, что балка, нагруженная на своем свободном конце сосредоточенным моментом $M = const$, будет балкой равного сопротивления изгибу, если ее сечение постоянно и диаметр определяется формулой $r = 3\sqrt{\frac{4M}{pa_1}} = const$.

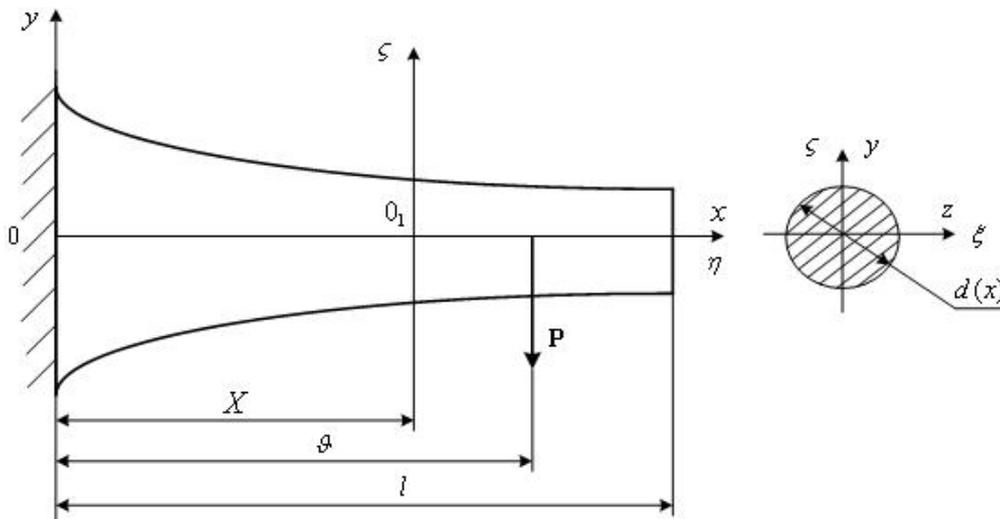


Рис. 1. К определению консольной балки равного сопротивления изгибу

2. Нагружение сосредоточенной силой

Если консольная балка равного сопротивления изгибу нагружена сосредоточенной силой, то из выражения (1) следует, что ее радиус определяется формулой

$$r(X) \approx 3\sqrt{\frac{4P(l-X)}{pa_1}},$$
(2)

где l - длина балки. Зависимость (2) иллюстрирует рис. 2.

3. Нагружение силой веса балки

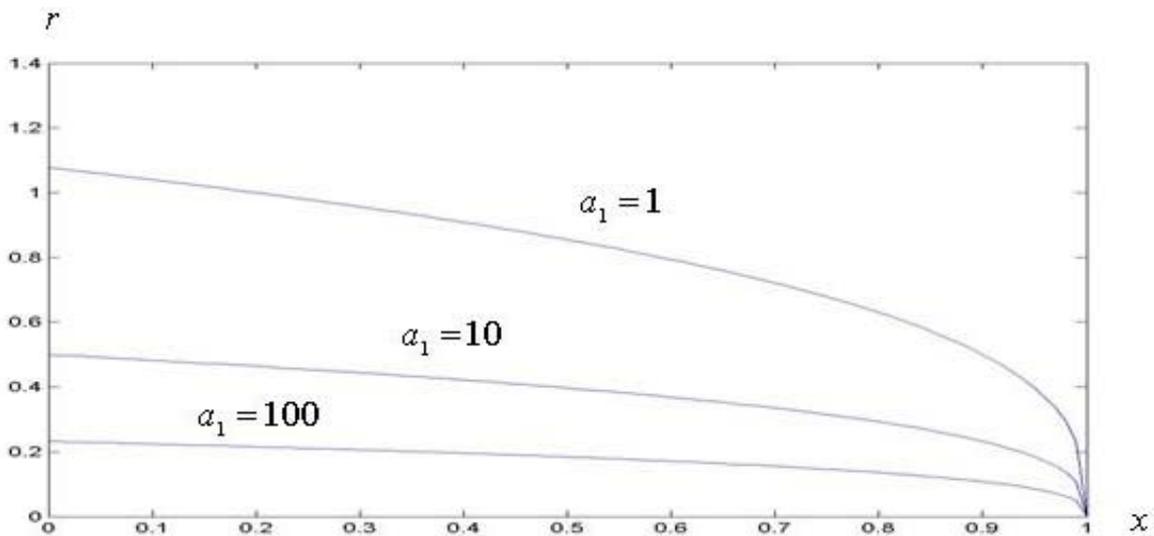
Рассмотрим консольную балку равного сопротивления изгибу при нагружении ее силой веса балки. Изгибающий момент силы тяжести P в сечении X равен

$$M_o(X) = P(X)(l - X), \quad (3)$$

где (X) - расстояние центра тяжести части балки, расположенной справа от сечения X , до начала координат,

$$P(X) = c \int_X^l S(x) dx = pc \int_X^l r^2(x) dx,$$

где $S(x) = pr^2(x)$ - площадь поперечного сечения балки.



$$l = 1.0; P = 1$$

Рис. 2. Форма образующей консольной балки равного сопротивления изгибу при нагружении сосредоточенной силой

Аналогично, статический момент рассматриваемой части балки относительно плоскости O_1 равен

$$M_{mo}(X) = \int_X^l xS(x) dx = p \int_X^l xr^2(x) dx.$$

Поскольку, хорошо известно,

$$(X) = \frac{M_{mo}(X)}{P(X)}$$

из выражения (3) следует, что

$$M_o(X) = p \int_X^l xr^2(x) dx - pcX \int_X^l r^2(x) dx. \quad (4)$$

Таким образом, из формул (1), (4) вытекает следующее интегральное уравнение относительно $r(X)$:

$$\frac{r^3(X)}{4} a_1 = \int_x^l x r^2(x) dx - cX \int_x^l r^2(x) dx. \quad (5)$$

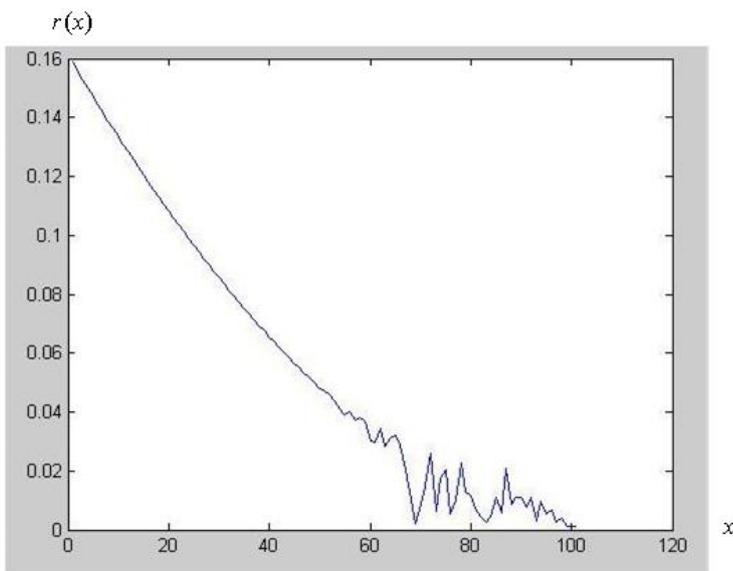
Утверждение 1. Если точное решение задачи (5) существует, то в нормах метрических пространств $L_1[0, l]$, $C[0, l]$ эта задача является некорректно поставленной [2].

Из утверждения 1 следует, что для получения корректного решения задачи (5) необходимо использовать тот или иной стабилизирующий функционал [2]. Логичным является построение стабилизирующих функционалов на основе требования гладкости решения задачи (5). Поэтому в качестве стабилизирующих функционалов рассматриваются

$$\text{функционалы вида } \Omega_1(r) = \int_0^l (r'(x))^2 dx, \quad \Omega_2(r) = \int_0^l (r''(x))^2 dx.$$

При численном решении задачи, приближенные значения функционалов $\Omega_1(r)$, $\Omega_2(r)$ могут быть вычислены путем замены производных соответствующими конечными разностями.

Поскольку аналитическое решение уравнения (5) не удается, рассматривается один из возможных численных методов решения этого уравнения. В этом методе задача сводится к задаче многомерной глобальной условной оптимизации, приближенное решение которой находится путем сведения ее к задаче многомерной глобальной безусловной оптимизации с помощью метода штрафных функций. Для решения последней задачи используется метод Нелдера-Мида в комбинации с методом мултистарта.



Некорректность задачи (5) иллюстрирует рис. 3, полученный без использования стабилизирующего функционала. Рисунку соответствует невязка решения, равная $\sim 1.4 \cdot 10^{-12}$.

$$l = 1.0; \quad c = 1.0; \quad a_1 = 1$$

Рис. 3. Форма образующей консольной балки равного сопротивления изгибу при нагружении силой веса балки: стабилизирующий функционал не используется

Литература

1. Глазунов В.А., Колискор А.Ш., Крайнев А.Ф. Пространственные механизмы параллельной структуры.- М.: Наука, 1991.- 94 с.
2. Каганов Ю.Т., Карпенко А.П. Математическое моделирование кинематики и динамики робота-манипулятора типа «хобот». // Наука и образование: электронное научно-техническое издание, 2009, 10, (<http://technomag.edu.ru/doc/133262.html>).

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ОБРАБОТКИ НЕЧЕТКИХ ЗАПРОСОВ В РЕЛЯЦИОННЫХ СУБД.

Белобородова М.Н.

Научный руководитель: к.т.н, доцент Волосатова Т.М.

МГТУ им. Н.Э.Баумана, Кафедра «САПР» (РК6), Москва, Россия

SOFTWARE FOR HANDLING FUZZY QUERIES IN RELATIONAL DATABASE SYSTEMS.

Beloborodova M.N.

Scientific chief: PhD, associate professor Volosatova T.M.

CAD/CAE Department (RK6) of BMSTU, Moscow, Russia

Аннотация

Применение аппарата теории нечетких систем в задачах поиска информации. Решение проблемы математического описания плохо формализуемой, слабо структурированной информации и ее обработка. Разработка приложения, которое осуществляет обработку нечеткого запроса, вывод его результата и итоговой степени принадлежности к нечеткому множеству.

Abstract

Use fuzzy logic to expressing the intent of a database query when the semantics of the query are rather vague. Development software for handling fuzzy queries in relational database systems.

Большая часть данных, обрабатываемых в современных информационных системах, носят четкий, числовой характер. Однако в запросах к реляционным базам данных, которые пытается формулировать человек, часто присутствуют неточности и неопределенности. Поэтому нечеткие запросы к базам данных - это перспективное направление в современных системах обработки информации.

Так как информация в базе данных (БД) хранится в четком виде, нося однозначный характер, то в нечетком запросе допустимо использовать только те нечеткие величины, которые в дальнейшем можно представить в виде числа или диапазона чисел.

Из классической теории множеств известно, что степень принадлежности (СП) элемента множеству может принимать два дискретных значения: 0 либо 1, т.е. либо строго принадлежать множеству, либо строго не принадлежать. Л.Заде[1] расширил классическое понятие множества, допустив, что СП может принимать любые значения в интервале [0; 1], а не только значения 0 либо 1. Такие множества были названы им *нечеткими*. Часто для удобства СП элемента множеству определяют по значению функции принадлежности (ФП), задаваемой аналитической формулой.

Существует свыше десятка типовых форм кривых для задания функций принадлежности. Наибольшее распространение получили: треугольная, трапецеидальная и гауссова функции принадлежности.

Треугольная функция принадлежности определяется тройкой чисел (a,b,c), и ее значение в точке x вычисляется согласно выражению:

$$MF(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b \\ 1 - \frac{x-c}{c-b}, & b \leq x \leq c \\ 0, & \text{в остальных случаях} \end{cases}$$

При $(b-a)=(c-b)$ имеем случай симметричной треугольной функции принадлежности, которая может быть однозначно задана двумя параметрами из тройки (a,b,c) .

Аналогично для задания трапецеидальной функции принадлежности необходима четверка чисел (a,b,c,d) :

$$MF(x) = \begin{cases} 1 - \frac{b-x}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ 1 - \frac{x-c}{d-c}, & c \leq x \leq d \\ 0, & \text{в остальных случаях} \end{cases}$$

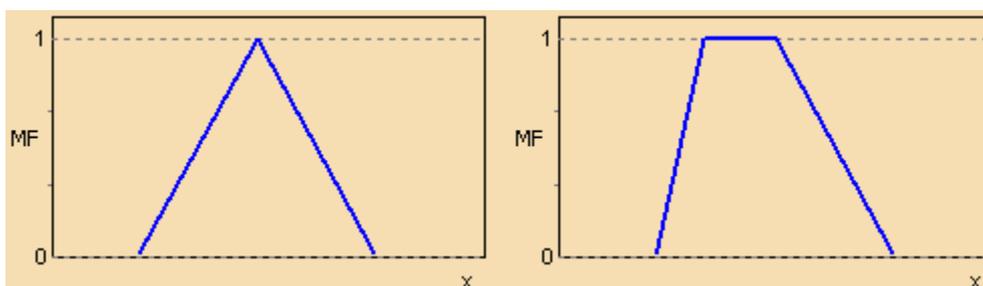


Рис.1. Типовые [кусочно-линейные функции](#) принадлежности.

Функция принадлежности гауссова типа описывается формулой

$$MF(x) = \exp\left[-\left(\frac{x-c}{\sigma}\right)^2\right]$$

и оперирует двумя параметрами. Параметр c обозначает центр нечеткого множества, а параметр отвечает за крутизну функции.

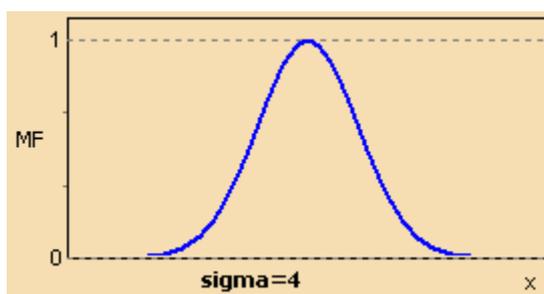


Рис.2. Гауссова функция принадлежности.

Представим ситуацию — имеется таблица в БД с информацией о габаритных размерах автомобилей. И нам необходимо выбрать автомобиль довольно высокий и недлинный. Нашему требованию отвечает машина, высота которой составляет примерно 1500 м, а длина - 4600м.

| | id | brand machine | height | length | width | baggage volume |
|--------------------------|----|----------------------|--------|--------|-------|----------------|
| <input type="checkbox"/> | 1 | Toyota Corolla | 1470 | 4540 | 1760 | 450 |
| <input type="checkbox"/> | 2 | Toyota Auris | 1515 | 4220 | 1760 | 354 |
| <input type="checkbox"/> | 3 | Toyota Avensis | 1480 | 4645 | 1760 | 520 |
| <input type="checkbox"/> | 4 | Toyota Camry | 1480 | 4815 | 1820 | 535 |
| <input type="checkbox"/> | 5 | Toyota Land Cruiser | 1950 | 4950 | 1970 | 590 |
| <input type="checkbox"/> | 6 | Toyota Rav4 | 1720 | 4395 | 1815 | 586 |
| <input type="checkbox"/> | 7 | Toyota Yaris | 1530 | 3750 | 1695 | 272 |
| <input type="checkbox"/> | 8 | Peugeot Boxer 2.2 MT | 2761 | 5998 | 2050 | 17000 |
| <input type="checkbox"/> | 9 | Lexus RX 350 | 1715 | 4740 | 1845 | 439 |
| <input type="checkbox"/> | 10 | Lexus GS 300/430 | 1430 | 4820 | 1820 | 430 |
| <input type="checkbox"/> | 11 | Lexus RX 400 H | 1735 | 4760 | 1845 | 439 |
| <input type="checkbox"/> | 12 | Lexus LX 570 | 1920 | 4990 | 1970 | 460 |
| <input type="checkbox"/> | 13 | Lexus SC 430 | 1370 | 4535 | 1825 | 135 |
| <input type="checkbox"/> | 14 | Audi TT Coupe | 1352 | 4178 | 1842 | 290 |
| <input type="checkbox"/> | 15 | Audi Q7 | 1737 | 5086 | 1986 | 775 |
| <input type="checkbox"/> | 16 | Audi A4 | 1427 | 4586 | 1772 | 460 |
| <input type="checkbox"/> | 17 | Audi A3 | 1421 | 4203 | 1765 | 281 |
| <input type="checkbox"/> | 18 | Audi R8 | 1252 | 4431 | 1904 | 100 |
| <input type="checkbox"/> | 19 | Peugeot 308 | 1498 | 4276 | 1815 | 348 |
| <input type="checkbox"/> | 20 | Peugeot 206 sedan | 1452 | 4188 | 1810 | 340 |
| <input type="checkbox"/> | 21 | Peugeot 107 | 1470 | 3430 | 1630 | 139 |

Рис.3. База данных автомобилей.

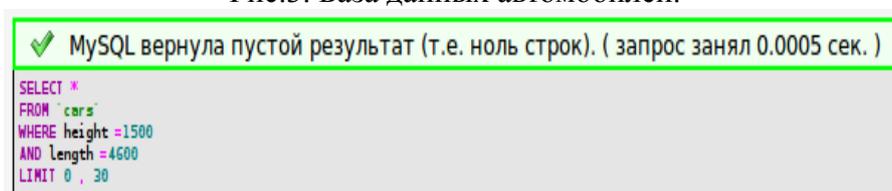


Рис.4. Результат четкого запроса к БД.

Это означает, что в мире нет ничего идеального. И человек не может точно определить, какими габаритами должен обладать довольно высокий недлинный автомобиль, чтобы такой автомобиль непременно был в базе. Поэтому мы будем отталкиваться от того, что в нашем представлении автомобиль, имеющий высоту примерно 1500, а длину примерно 4600, считается довольно высоким и недлинным. Можно, конечно, задать запрос, используя некий интервал значений, в этом случае мы получим результат, но он будет слишком ограничен. Ведь там не будет тех автомобилей, которые отвечают первому условию и совсем немного не соответствуют второму или наоборот. В итоге мы подошли к тому, что в нашем случае без использования нечеткой логики не обойтись.

Предлагается представить нужные габариты автомобиля высоту и длину, как лингвистические переменные, которые могут принимать значения:

Высота = {низкая, высокая}

Длина = {длинная, короткая}

Тогда SQL-запрос примет вид

Select * From cars

Where height=fairly high

AND length= rather short;

Для выполнения нечеткого запроса необходимо каким-либо образом преобразовать его в четкий запрос, соответствующий стандарту. Но сначала необходимо описать и задать непосредственно нечеткое множество, которое, как известно, характеризуется:

- диапазоном возможных значений;
- типом функции принадлежности;
- параметрами функции принадлежности.

Также для запроса необходимо задать некий *порог* (число, *степень доверия*), для получения элементов, степень принадлежности которых не ниже указанного числа — α -уровня.

Опишем и зададим нечеткое множество Довольно высокая машина:

- 1) диапазон возможных значений: 1000÷3000 (как правило, в такой диапазон входит возможная высота легкового автомобиля);
- 2) тип ФП: обобщенная колокообразная (gbell);
- 3) параметры ФП: $a=1500$, $b=100$, $c=4$;
- 4) α -уровень: 0.9;

Аналогично и для нечеткого множества Недлинный автомобиль:

- диапазон возможных значений: 2000:6000;
- тип ФП: обобщенная колокообразная (gbell);
- параметры ФП: $a=4600$, $b=100$, $c=4$;
- α -уровень: 0.9;

Осталось сформировать итоговую степень принадлежности элемента к нечеткому множеству. Итоговую степень принадлежности назовем $qcix$.

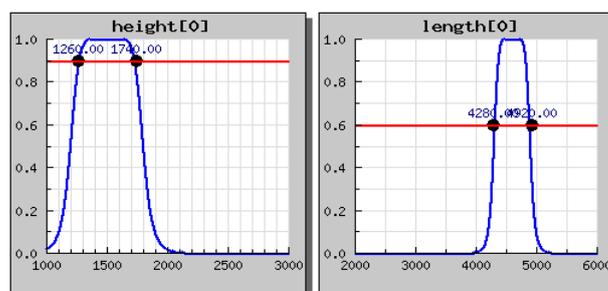
Для нашего запроса:

$$\mu(X) = \min(\mu_{дл}(X), \mu_{вс}(X))$$

где: $\mu(X)$ - $qcix$; $\mu_{дл}(X)$ - СП к множеству "Довольно высокая машина"

$\mu_{вс}(X)$ - СП к множеству "Недлинный автомобиль".

Schedule of membership function



Query Result

| id | brand machine | height | length | width | baggage volume | qcix |
|----|------------------|--------|--------|-------|----------------|------|
| 1 | Toyota Corolla | 1470 | 4540 | 1760 | 450 | 1 |
| 3 | Toyota Avensis | 1480 | 4645 | 1760 | 520 | 1 |
| 13 | Lexus SC 430 | 1370 | 4535 | 1825 | 135 | 1 |
| 16 | Audi A4 | 1427 | 4586 | 1772 | 460 | 1 |
| 4 | Toyota Camry | 1480 | 4815 | 1820 | 535 | 0.93 |
| 9 | Lexus RX 350 | 1715 | 4740 | 1845 | 439 | 0.93 |
| 6 | Toyota Rav4 | 1720 | 4395 | 1815 | 586 | 0.92 |
| 10 | Lexus GS 300/430 | 1430 | 4820 | 1820 | 430 | 0.92 |
| 11 | Lexus RX 400 H | 1735 | 4760 | 1845 | 439 | 0.88 |

Литература

1. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. – М.: Мир, 1976.
2. Масалович А. Нечеткая логика в бизнесе и финансах. www.tora-centre.ru/library/fuzzy/fuzzy-.htm
3. С.Д.Штовба "Введение в теорию нечетких множеств и нечеткую логику" <http://matlab.exponenta.ru/fuzzylogic/book1/index.html>.

СЕГМЕНТАЦИЯ ТЕКСТОВЫХ ИЗОБРАЖЕНИЙ

Бодренко Е.А.

Научный руководитель: к.т.н., доцент каф. РК6 Волосатова Т.М.
МГТУ им. Н.Э.Баумана, Кафедра «САПР» (РК6), Москва, Россия

SEGMENTATION OF TEXT IMAGES

Bodrenko E.A.

The scientific chief: Ph.D. Volosatova T.M.
Bauman Moscow State Technical University,
Department RK6 of BMSTU, Moscow, Russia

Аннотация

В докладе представлен анализ и метод решения задачи сегментации текстовых изображений и кратко представлен обзор методов сегментации изображений и различные аспекты их применения. Представлено решение задачи сегментации текстовых изображений на основе анализа значения яркости пикселей изображения.

Abstract

This report is about analysis and a method of decision of a problem of segmentation of text images and the review of methods of segmentation of images and various aspects of their application is short presented. The decision of a problem of segmentation of text images on the basis of the analysis value of brightness of pixels of the image is presented.

Введение

Сегментация обычно понимается как процесс поиска однородных областей в изображении. Этот этап весьма трудный и в общем виде не алгоритмизированный до конца для произвольных изображений. Наиболее распространены методы сегментации, основанные на определении однородных яркостей (цветов) или однородностей типа текстур.

При существовании стабильных различий в яркостях отдельных областей поля зрения применяются пороговые методы. Методы наращивания областей эффективны при наличии устойчивой связности внутри отдельных сегментов. Метод выделения границ хорошо применять, если границы достаточно четкие и стабильные. Перечисленные методы служат для выделения сегментов по критерию однородных яркостей. Заметим, что один из самых эффективных методов наращивания областей предполагает выбор стартовых точек либо с помощью оператора (алгоритм центроидного связывания), либо автоматически.

Все методы весьма приемлемы с точки зрения вычислительных затрат, однако, для каждого из них характерна неоднозначность разметки точек в реальных ситуациях из-за необходимости применения эвристик (выбор порогов совпадения яркостей, выбор цифровых масок и т.д.). Заслуживает внимания в связи с этим предложенный метод многозначной разметки, основанный на комбинации различных приемов для снижения неопределенности. Важное практическое значение имеют допускающие параллельную обработку алгоритмы ускорения процесса разметки на основе логического анализа соседних элементов.

Этапы сегментации текстовых изображений

Под распознаванием документов обычно подразумевают процесс преобразования информативного содержания документа из бумажного варианта в электронный вариант с сохранением логического содержания и, по возможности, форматирования. Сегментация – это процесс идентификации текстовых областей, картинок, таблиц, математических формул, а также определение порядка их следования или взаимного расположения. Под сегментацией понимается взаимодействие геометрической и логической сегментации. Геометрическая сегментация занимается выделением геометрической структуры документа. Этот этап включает в себя следующие шаги: предобработка (preprocessing step) и разбиение (page decomposition). Последний шаг – это разбиение образа документа на максимальные области,

которые относятся по содержанию к различным типам. К предобработке обычно относят бинаризацию и нахождение угла наклона (skew) страницы. Под логической сегментацией понимается поиск различных логических интерпретаций для найденных областей (заголовков, параграф, подпись и т.п.) и отношений между областями (подпись к картинке, колонки, фреймы и т.п.).

Выделение разделителей

В данном докладе используются образы документов, в которых информационное заполнение черное, а фон белый. Для алгоритма разбиения требуется структурированная информация о свободных областях. Существуют разные способы их представления и анализа. Будем считать, что распознаваемый документ уже прошел процесс предобработки, а именно была выполнена бинаризация и достигнута правильная ориентация текста.

Исходное изображение можно представить как матрицу нормированных яркостей точек $B: B = \{b_{ij}\}, 0 \leq b_{ij} \leq b_{max}, i=1 \dots n; j=1 \dots m$, где n - ширина картинки, m - высота картинки.

Для определённости будем считать, что максимальное значение нормированной яркости (b_{max}) соответствует чёрному цвету а минимальное (равное 0) - белому.

Горизонтальные полосы

Под горизонтальной полосой будем понимать прямоугольник. У нее могут быть уточняющие от одной вертикали до другой вертикали прямоугольники, которые расположены по оси Y на расстоянии, не превышающем заданную константу y , как изображено на [рисунке 1](#).

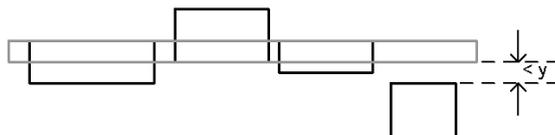


Рис.1. Горизонтальная полоса

Для выделения горизонталей используется алгоритм выделения горизонтальных линий от края до края. На вход алгоритму подается набор компонент и интересующая область. На выходе изображение “разрезано” на горизонтальные линии, которые определяются прямоугольниками. Задача выделения строк сводится к нахождению верхних и нижних граней строк текста, изображённого на исходном изображении.

Алгоритм сегментации строк основывается на том, что среднее значение нормированной яркости в изображениях межстрочных промежутках существенно ниже, чем в изображениях текстовых строк.

Сначала для всех пиксельных строк исходного изображения находим их средние

значения нормированной яркости $s_j = s_j(B) = \frac{1}{n} \cdot \sum_{i=1}^n b_{ij}$ и определяем среднее значение

$$s(B) = \frac{1}{m} \cdot \sum_{j=1}^m s_j(B)$$

нормированной яркости всего изображения

Среднее значение нормированной яркости в межстрочных промежутках текста должна быть невелико (в идеальном случае равно нулю). Поэтому нормированную яркость верхней границы текстовой строки можно выразить через среднее значение нормированной яркости изображения $st=kt*s(B)$ где $0 < kt < 1$ – коэффициент. Аналогично для нижней границы текстовой строки.

Работа алгоритма сегментации строк заключается в последовательном просмотре массива средних значений и выявлении множества пар индексов пиксельных строк, соответствующих верхней и нижней граням изображения строки. Таким образом, в результате работы алгоритма на исходном изображении отмечается положение всех текстовых строк.

Для определения строчности найдем горизонтали для всей исходной области. Далее поделим область на N частей по вертикали и для каждой части найдем горизонтали. Теперь остается проанализировать полученную информацию. Если горизонтали в частях области не противоречат горизонталям найденным во всей области, то будем считать, документ строчным. Например, как показано на рисунке 2, если были найдены новые горизонтали, то область документа не строчная.



Рис.2. Пример не строчного документа

Вертикальные полосы

На следующем этапе решения задачи сегментации изображения текста, из изображений строк. Входом для алгоритма сегментации слов служит изображение, какой-либо одной текстовой строки, которое получается из исходного изображения документа после разбиения его на горизонтальные полосы.

Алгоритм сегментации слов основывается на том, что среднее значение нормированной яркости в межсловных интервалах существенно ниже, чем в изображениях слов. Он похож на алгоритм сегментации строк, только просмотр идет по пиксельным столбцам изображения строки.

Для улучшения качества работы алгоритма выделения слов из строки вначале его работы повышается контрастность строки и применяется “размазывающий фильтр”, результаты которого представлены на рисунке 3.

Полученный результат. Вид в начальной момент времени и в конечном момент времени.

Рис.5. Результат работы "размазывающего" фильтра.

Для всех пиксельных столбцов исходного изображения строки находим их средние

$$c_i = c_i(B) = \frac{1}{m} \cdot \sum_{j=1}^m b_{ij}$$

значения нормированной яркости. Затем определяем среднее значение

$$c(B) = \frac{1}{n} \cdot \sum_{i=1}^n c_i(B)$$

нормированной яркости для данного изображения строки, где m и n, - высота и ширина текущей строки в точках соответственно.

Среднее значение нормированной яркости в межсловных интервалах должно быть невелико. Поэтому левую границу (начало слова) можно выразить через среднее значение нормированной яркости изображения строки $cl = kl \cdot c(B)$, где $0 < kl < 1$ – коэффициент. Аналогично для правой границы (конец слова).

В большинстве изображений слов символы расположены близко друг к другу и межсимвольные интервалы не так ярко выражены, как в случае межстрочных или межсловных интервалов. Поэтому алгоритм сегментации символов сложнее алгоритмов сегментации строк и слов. Необходимо выявлять и удалять ложные границы символов, которые разрезают символ на части. Это может происходить с буквами П, Н, Ц. Причём, символ может быть связан, либо в верхней (П), либо в средней (Н), либо в нижней части (Ц)

пиксельных столбцов. Необходимо разделить изображение на три уровня по вертикали и анализировать эти уровни отдельно друг от друга.

В результате получаем изображение текста, каждый символ которого отделен от остальных символов, то есть сегментация текстового изображения достигнута.

Заключение

В последние годы появляется все больше прикладных областей, в которых актуально применение методов сегментации изображений — это системы распознавания документов, мультимедийные базы данных, цифровая фотография, сжатие изображений и другие.

Сегментация является, как правило, первым этапом в работе систем анализа изображений и, возможно, самым важным и сложным. Все последующие этапы анализа изображений, такие как извлечение признаков или распознавание объектов, напрямую зависят от качества сегментации.

Литература

1. *Гонсалес, Р.* Цифровая обработка изображений – М. : Техносфера, 2005.
2. Балыков, Е.А. Разработка и обеспечение качества программных средств обработки и анализа изображений / Е.А. Балыков, В.А. Царев – М.: Компания Спутник+, 2006.
3. Путятин Е.П., Аверин С.И. Обработка изображений в робототехнике. М: Машиностроение, 1990.
4. Ту Дж., Гонсалес Р. Принципы распознавания образов / пер. с англ. – М. : Мир, 1978.
5. Интернет-ресурс ru.wikipedia.org.

BGP ROUTING TABLE: TRENDS AND CHALLENGES

Alexander Afanasyev, Neil Tilley, Brent Longstaff, and Lixia Zhang

Computer Science Department, University of California,
Los Angeles

{afanasev,tilleyns,blongsta,lixia}@cs.ucla.edu

Abstract

BGP is currently the most important protocol for ensuring global connectivity over the Internet. This puts a great deal of responsibility on BGP and creates a number of challenges for it. Of primary concern is the impact that various currently deployed BGP-based techniques have on the scalability of the global routing table. While these techniques provide ISPs with additional traffic management services (e.g., balancing, multi-homing, etc.), they have expanded the routing table size at a pace that exceeds the allocation rate and is increasing. In our study we present a two-level analysis of BGP announcements for the period 2003–2009. First, we correlate IP allocation data with globally announced prefixes and show how efficiently ISPs announce their allocated address space. Second, we correlate BGP announcement data to itself and show various internal factors that contribute to routing table growth. Finally, we document in which regions of the world routing announcements have originated during the period of this study, and we draw conclusions about the spread of global Internet connectivity.

1 Introduction

BGP (Border Gateway Protocol) [1] is a key component enabling Internet routing worldwide. The routing protocol operates at the junction where independent networks (ASes, or autonomous systems) exchange network traffic to ensure global connectivity. Because ASes are separate networking and economic entities, BGP currently operates while essentially balancing two purposes that are for the most part orthogonal to one another. First, it interconnects all ASes in the world. Second, BGP tries to satisfy a wide variety of ISP-specific routing policies, which are governed by operating costs, a number of agreement-based and politically-based issues, network locality, multi-homing preferences, and, in some select cases, traffic connection capacity.

The routing table has expanded enormously over the past ten years due to fractionalization and finer segmentation of the IP address space. The table now maintains more entries than a hierarchical structure would have yielded that had worked with strictly consolidated blocks. Fig. 1 shows that over the past six years, the number of the global routing table entries has more than doubled. IP address allocations have also doubled in the same period, but numerically, all new allocated blocks account for less than 18% of the actual entries in the BGP routing table ($\approx 50k$ new allocations from 2003 to 2009, compared to $\approx 300k$ entries in the routing table in 2009).

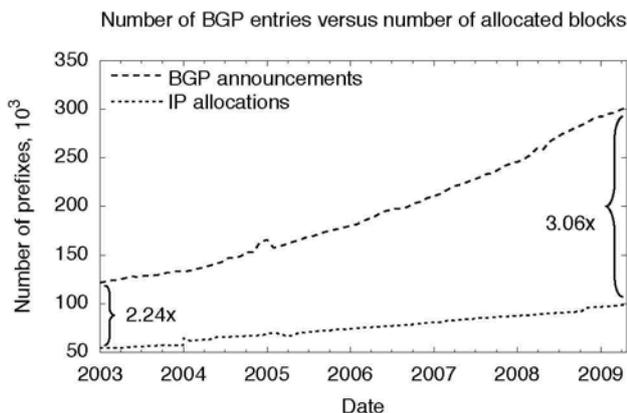


Fig. 1. Number of BGP entries compared to number of allocated IP blocks table

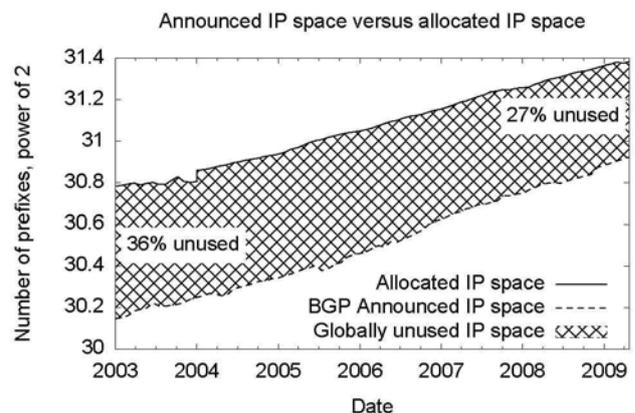


Fig. 2. Comparing the IP space that has been allocated to the amount of IP space announced in the BGP

There is an interesting aspect to notice with the BGP table dynamics. While the routing table has experienced substantial growth in comparison to the number of new IP blocks allocated, all the IP space covered by the entries in the routing table is still only a fraction of the total measured IP space

that has been allocated. This is shown in Fig. 2, which illustrates that over time, the amount of dormant IP space—allocated, but not announced in routing tables—has ranged from a little over 1/3 to a little over 1/4 of the total. Despite the high ratio between the number of announced prefixes in the BGP table and the number of actually allocated IP blocks (2.24x in 2003, 3.06x in 2009), a significant portion of IP space ($\approx 27\%$) remains unreachable globally.

In this paper we present an extensive analysis of the IP allocation and BGP announcement statistics. First, we examine the dynamics of IP address allocation and recent history regarding which prefix block sizes are most popular to allocate to ISPs (Internet Service Providers). Second, we examine the correlation between the global routing table and IP prefix allocation data. Among the common prefix block sizes allocated, we show what size blocks are most common in the BGP table. Third, we present a summary by geography of those regions around the globe that contribute to the BGP table contents each year. We also identify the regions where the most rapid development of ISP hosting has occurred. Fourth, we estimate the lifespan of BGP entries as measured between 2003 and 2009. We then discuss a number of factors that contribute to the marked growth in size of the routing table.

The rest of the paper is organized as follows. Section 2 presents an analysis for IP address allocation statistics. In Section 3 we analyze the composition of the BGP table and its changes over time, as well as the stability of routing table entries. Finally, we present related work and conclusions in Sections 4 and 5, respectively.

2 IP address allocation dynamics

2.1 Allocated IP block sizes

The collected data highlight a number of aspects about allocated block sizes over the past six years. Fig. 3 illustrates several notable changes. It shows the number of allocations increasing for every prefix length in every year, though at different rates. In all years, clearly /24 prefix allocations are the most prevalent. At the same time, there has been almost no increase in /16 blocks. Overall, this shows the growing trend toward smaller blocks as the IPv4 address space approaches saturation.

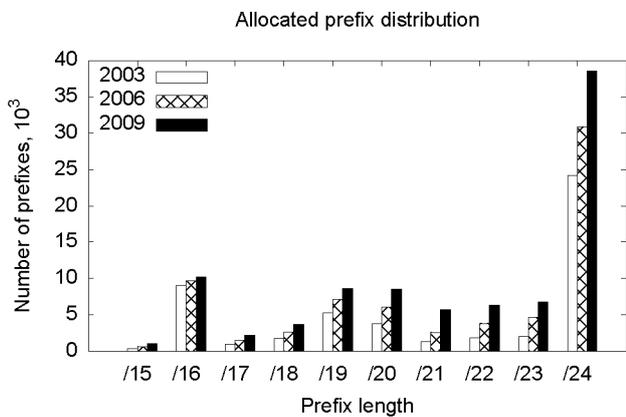


Fig.3. Allocated prefix distribution

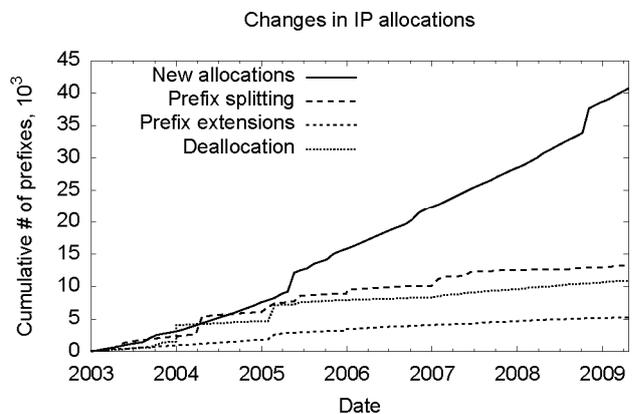


Fig.4. Changes in IP allocations

2.2 Yearly distribution of IP allocations

Fig. 4 charts the occurrence of other events that factor into the net change of the allocation count. These include prefix splitting, prefix extension, and deallocation. Prefix splitting is the dividing of one prefix into multiple smaller prefixes (e.g., a /16 prefix can be split into two /17). Prefix extension is the aggregating of an existing prefix with its adjacent, previously unallocated address space (e.g. a /17 prefix might become a /16 prefix by including the adjacent /17 address space). Deallocation is the withdrawing of a prefix allocation, essentially releasing IP space for later use.

The number of new allocations over the studied time period exceeds 40k. A smaller amount (13k) is due to prefix splitting. A small decrease in the allocations count (5k) has resulted from prefix extension. Finally, a larger decrease is due to deallocation (10k), but the decrease is nevertheless greatly outpaced by the number of new allocations. To summarize, growth in the

allocation count is due primarily to new allocations. Steady, quasi-linear growth in recent years suggests that the increase in allocations will continue at a similar, moderate pace after IPv6 deployment.

3 BGP routing table

3.1 Analysis of BGP table growth factors

The BGP routing table is growing at a rate significantly higher than the pace that RIRs (Regional Internet Registries) are allocating IP blocks. Across all observed BGP monitors [2, 3], the average number of entries in the global routing table is more than 3 times the number of IP blocks that RIRs have allocated (refer Fig. 1). This multiplication in size reflects two primary practices. First, ISPs tend to subdivide allocated IP blocks into several individual prefixes and announce them separately. Such behavior is typical among transnational providers as well as among ISP customers that have been lent parts of their service providers' address space and, in turn, independently announce subdivided IP address blocks. Second, various traffic engineering techniques (traffic balancing, multihoming, etc.) give rise to situations where the same address block is covered by several announced prefixes.

IP block fragmentation. The contents of the BGP routing table consist of IP prefixes that either match, fragment, or aggregate various IP allocation blocks. Fig. 5 shows the correlation between allocated IP blocks and announced IP prefixes and the relative proportions of these three categories over time. The matched curve in the figure represents IP blocks that are announced in the routing table in the exact form that they were issued by RIRs. As evident in the figure, the number of matched prefixes accounts for 1/6 of the total BGP entries presently, with the trend that this fraction is growing smaller over time.

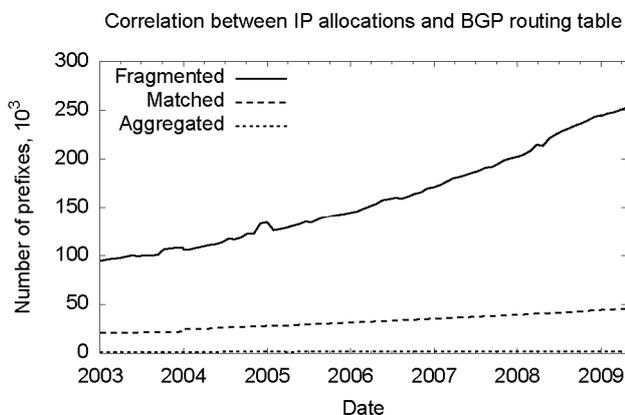


Fig. 5. Dynamics of matched, fragmented, and aggregated IP prefixes in BGP announcements

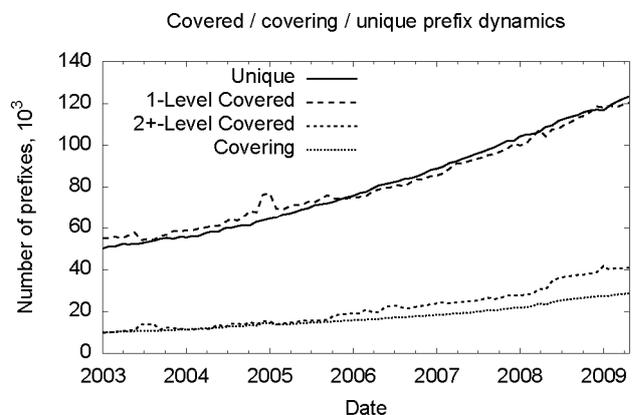


Fig. 6. Relative numbers of covered, covering, and unique IP prefixes among BGP announcements

ISPs are evidently not inclined to use address space in the form in which it has been allocated. For various possible reasons (e.g., geographical dispersion), ISPs split up allocated blocks into a number of sub-blocks and announce each of these independently. The fragmented curve in Fig. 5 represents these subdivided blocks, which account for more than 83% of all entries in the global routing table. IP block fragmentation poses one of the primary concerns for future scalability of the BGP routing table.

The lowest curve, aggregated, represents IP prefix announcements that cover several allocated IP blocks. For these cases—in contrast to fragmentation—ISPs that have several adjacent IP block allocations simply announce them as a single IP prefix. As the figure illustrates, this aggregation technique is rarely employed in any measure. Its primary intent, to reduce the number of entries in the routing table, is outweighed far more by other policy routing choices. The number of aggregated prefixes in 2003 was 1,400. By 2009 this number increased only marginally, to just under 2,000

prefixes, accounting for under 1% of the entire routing table and thus is considered negligible.

This observed behavior has a measure of relevance to future IPv6 deployment. ISPs tend not to announce their allocated IP spaces in their original form. This behavior occurs regardless of the size of IP block that an ISP has been allocated. According to current RIR policy, the minimum allocation for an IPv6 block is /32 [4]. In terms of cost, the price for an IPv6 /32 block is the same as for a /19 or /20 IPv4 address block [5], meaning for an equivalent amount to obtain fewer than 10,000 IPv4 addresses, ISPs can be assigned an IPv6 block several orders of magnitude larger than the entire IPv4 space. If allocations of large blocks continue, it is likely to mitigate the problem of multiple, non-adjacent IP block allocations per customer. However, without a major change in the BGP protocol aimed at lessening incentives to announce fragmented IP prefixes, increasing the allocated IP block size will not significantly assist in reducing the size of global routing table. Table size reduction stems only from aggregatable and matching IP prefixes. In other words, only ISPs that currently use all allocated IP space as a single IP block (i.e., matched or aggregated) have any likelihood of using a bigger space provided in IPv6 also as a single block. We conclude that the upper bound of IP space announcement optimization is limited by the number of matched prefixes, which currently stands at less than 17% of all prefixes.

Duplicate announcements of IP blocks. The BGP routing table has a large number of prefix ranges that overlap each other. Such IP address coverage duplication assists in calculating an actual route by matching the destination address with the longest available prefix. Address duplication in a routing table is, in theory, an effective way to reduce the size of the routing table itself. Our observations indicate that ISPs use the fundamental IP routing feature of longest-prefix matching extensively. As Fig. 6 shows by the 1-level and unique curves, the number of IP prefixes in the BGP table which are covered by exactly one bigger prefix is nearly the same as the number of unique prefixes (i.e., base prefixes). There is moreover a substantial number of prefixes that have several layers of coverage (several duplication levels—refer to 2+-level curve). We also have found that about half of the autonomous systems ($\approx 58\%$ in 2003 and $\approx 44\%$ in 2009) employ prefix duplication. This finding proves that prefix duplication is common practice on the Internet.

These high proportions of 1-level and 2+-level covered prefixes indicate there are other incentives and benefits for prefix duplication, in addition to the theoretical routing table optimization. One factor we consider is a multi-provider connection for end-networks (so-called multi-homing of stub networks). As Oliveira et al. [6] have stated, more than 70% of all BGP announcements belong to multi-homed stub networks. In other words, the global routing table is employed to serve local or semi-local routing interests for most customers. Since these routing interests have importance primarily on a local scale, it is unlikely that the outside world follows widely divergent routing paths to reach various providers' connections to a multi-homed customer. While the need for covered prefixes is evident to accommodate various routing preferences, they need not be shared universally in routing tables. Accordingly, with counter-incentives to IP prefix fragmentation, a significant reduction in the size of the BGP routing table is conceivable. For example, deployment of geography-based techniques (e.g., GIRO protocol [7]) can assist in reducing the global routing table to 1/4 of the current size. As an area of further research, separate means for multi-homing and traffic engineering tasks might be provided.

3.2 Analysis of the BGP table contents

BGP announcements by geographical region. In examining the global routing table content, we have conducted a country-based analysis of the distribution of globally announced IP prefixes. Fixed-time snapshots point out the major contributors to the global routing table and give an understanding of the Internet's penetration globally. Country assignments for IP prefixes are taken from RIR data, which provide enough precision for a global-scale analysis.

By way of summarizing results, we present collected information for the six countries worldwide that have the most impact on the global routing table in 2009 (Table 1). Matching their 2003 and 2009 numbers side-by-side, they provide easy comparison of changes over time. All countries contribute in greater numbers to the BGP table. The United States retains its leading position. Meanwhile the

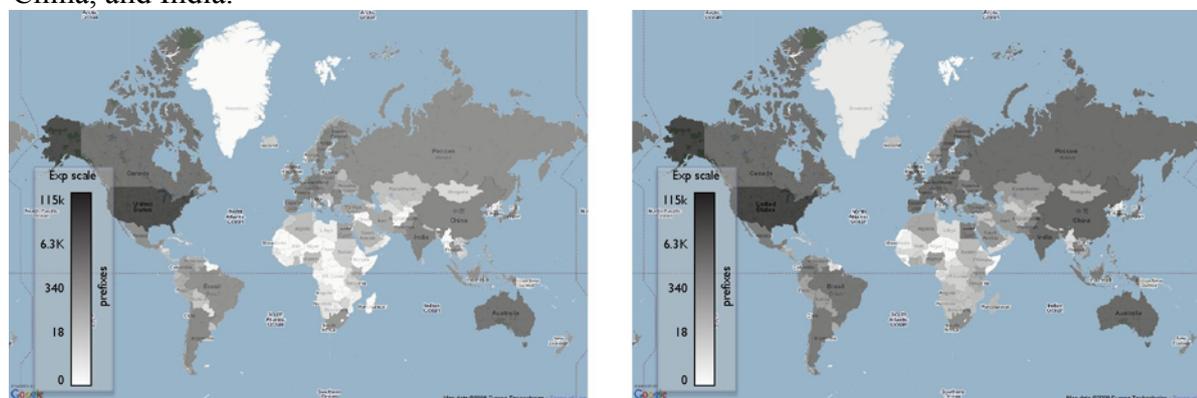
ordering of the rest of the contributors significantly changes. For example, South Korea becomes the second major contributor to the size of the global routing table, and China ranks third.

Table 1. Statistics for announced IP prefixes and corresponding IP space

| Country | Prefixes | | IP addresses | |
|-------------|----------|------|--------------|------|
| | | 2009 | 2003 | 2009 |
| US | 65 | 116 | 760 | 1170 |
| South Korea | k | k | M | M |
| Korea | 2k | 14k | 27M | 84M |
| China | 2k | 13k | 28M | 224M |
| Australia | 6k | 11k | 47M | 38M |

Comparing IP space usage to the number of announcements attributed to a given country, China, Japan, the European Union, Germany, and South Korea are responsible for the most amount of announced address space, behind the United States. These five geographic areas are a different ordering from that given in Table 1 (South Korea, China, Australia, India, Russia). This fact highlights that some countries announce a large number of relatively small prefixes (e.g., in South Korea one prefix covers on average 5,900 addresses), whereas some announce a small number of large prefixes (e.g., in Japan one prefix covers on average 37,200 addresses). If this difference in usage efficiency occurs because of additional government regulations, then for future IPv6 deployment, a similar host of regulations should be considered globally.

Fig. 10 shows a comparison of Internet penetration globally between years 2003 and 2009. Note that in both illustrations, shades of gray assigned to countries correspond to an exponential scale. Comparing 2009 with 2003, all regions show various degrees of greater Internet hosting. While the United States retains the lead position throughout the years measured, several regions with emerging economies exhibit marked change (toward darker shades) within the six-year span, such as Russia, China, and India.



(a) 2003

(b) 2009

Fig. 7. Geographical distribution of the number of announced prefixes. Given on an exponential color scale, from light (least) penetration to dark (greatest).

An extended presentation illustrating the geographical distribution of IP prefixes and IP space is available online, at <http://lasr.cs.ucla.edu/afanasyev/09-routing-map/>

Lengths of announced IP prefixes. Fig. 8 presents the distribution of announced prefix lengths, classified by each year. The majority of the global routing table entries are /24 prefixes and account for more than 53% of the entries. The number of /24 prefixes has nearly doubled between 2003 and 2009. At the same time, the number of blocks allocated that are actually /24 in size is 4 times smaller (refer Fig. 3). On the one hand, this is evidence that large numbers of stub networks (i.e., relatively small customer networks) use announcements of small address blocks to implement multi-provider connectivity. On the other hand, the excessive number of /24 prefixes poses a question about the relevance of current and future algorithms for IP space assignment (e.g., sequential,

bisection allocation, and GAP algorithm [8]). If the majority of ISPs tend to fragment an assigned address space into small chunks and announce them separately, attempts at minimizing the number of assigned IP blocks per organization will yield little effect and have only limited impact.

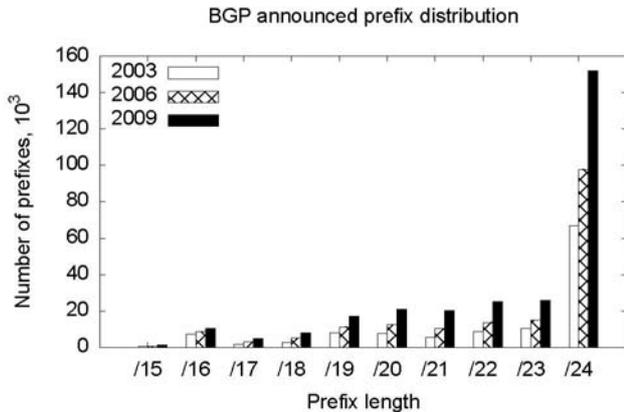


Fig. 8. Distribution of announced IP prefix lengths

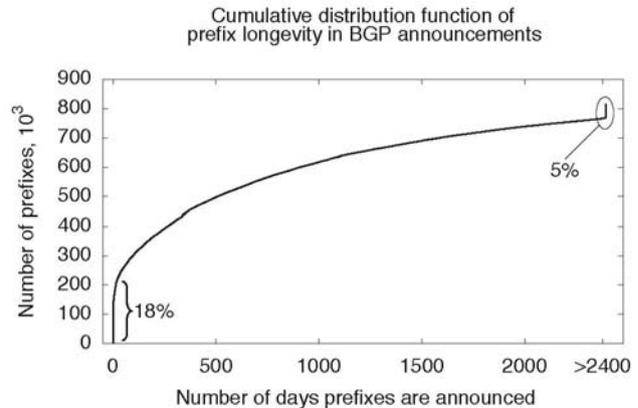


Fig. 9. Longevity of prefixes in BGP announcements

The results in this section indicate a tight relationship between global routing table growth and IP space fragmentation and duplication. If we could suppress the majority of non-globally related (i.e., locally concerned) announcements, such as bursts of /24 prefixes due to local traffic engineering, local and semi-local multihoming support, etc., the size of the global routing table could be significantly reduced.

Longevity distribution of BGP entries. Another aspect of the BGP announcement analysis is determining the stability of the global routing table. A cumulative distribution function of prefix longevity appears in Fig. 9. The step on the far right of the graph indicates that about 5% of all prefixes remain stable over the six-year period ($\approx 40k$ equal to 15% of the current global routing table). On the far-left side a large number of prefixes (18% of all prefixes observed) are active for only short periods of time (i.e., 1–3 days). One portion of these short-lived prefixes is likely composed of spammers that are known to hijack somebody’s (or even no one’s) prefix, announce it for a brief time, and send virtually untraceable spam messages [9]. Another portion can be attributed to configuration errors. The rest can be explained by normal BGP operations, where some prefix becomes briefly visible, as for example when a primary network channel malfunctions.

With regard to prefix length, we observe that /24 is prevalent both for long- and short-lived prefixes. However, most entries in the BGP table in the range from /25 to /32 are composed almost entirely of short-lived (unstable) prefixes.

Besides a fixed number of highly stable routes, more than half of the prefixes appear in the table for less than 500 days. It is unlikely then that a given route is visible for a very long time. This observation underscores that the composition of the routing table is highly dynamic and will pose challenges for future research.

4 Related work

Past studies [10–12] have characterized growth of the Border Gateway Protocol routing table in terms of the prevalence of special announcements to suit traffic engineering purposes. They also have measured the number of appearing and disappearing announcements in the BGP routing table, the latency between allocation and prefix appearance in BGP announcements, and the level of unallocated address announcements. Since the time of those studies (2003–2005), the BGP routing table has continued its growth. Our study examines the current state of the BGP routing table and quantifies how the high-level picture has changed from earlier measurements.

Geoff Huston’s Potaroo project [13] presents up-to-date measurements of the BGP routing table growth and IP allocation dynamics, dating back to 1994. However it does not analyze the impact

that fragmentation and address space duplication have on the BGP table growth over time and how they affect estimates of the routing table size in the future.

The contribution of our research is to document where routing announcements are originating around the world—not necessarily a measure of where most new Internet traffic is occurring, but a way to witness the spreading of Internet infrastructure connectivity around the globe. Coupled with an analysis of how long these announcements stay in the routing table—a measure of table stability—it is possible to make some projections of how the Internet continues to diversify.

5 Conclusions

We have analyzed BGP announcement snapshots provided by the University of Oregon Route Views and RIPE NCC Routing Information Service projects. Between 2003 and 2009, a span of six years, the average size of the BGP routing table has more than doubled. Likewise the IP address allocations have also doubled in the same period. Numerically, however, all the new allocated blocks have added less than 18% of the total entries in the BGP routing table.

We have identified several primary causes of the accelerated BGP routing table growth. They are as follows: First, fragmentation of allocated IP blocks; more than 80% of announced prefixes are from allocated IP blocks that have been subdivided. Second, announced space duplication; more than 54% of the address space is covered at least twice in the global routing table. This duplication highlights an emerging problematic trend of using the global routing table to serve local interests, e.g., to implement traffic engineering and multi-provider connections.

The content analysis of BGP routing table announcements shows that the majority of globally announced prefixes (>50%) are of /24 size. This strengthens the conclusion that a substantial number of entries in the global routing table serves local, not global, interests of small customer networks. The content of the global routing table is highly dynamic. Although there is a small portion of highly stable entries (<15%), the remainder of the BGP table content fits an exponential tapering-off distribution for prefix longevity.

Our examination of the geographical distribution of IP allocation and BGP announced prefixes shows a depth Internet penetration around the globe that is wide-ranging. We have observed a number of quasi-exponential distributions for various measurements, including for the following: the geographical distribution of the number of allocated prefixes, the numbers of the corresponding address spaces, the number of announced prefixes, and the corresponding globally announced address spaces. Moreover these distributions have not significantly changed in character over the last six years.

Appendix A Geographic representation of the distribution of IP prefixes

Figs. 10 and 11 show a comparison of Internet penetration globally between years 2003 and 2009. Note that in all illustrations shades of gray assigned to countries correspond to an exponential scale. Comparing 2009 with 2003, all regions show various degrees of greater Internet hosting. While the United States retains the lead position throughout all years measured, several regions with emerging economies exhibit marked change (towards darker shades) within the span of 6 years, such as China, Russia, and Viet Nam.

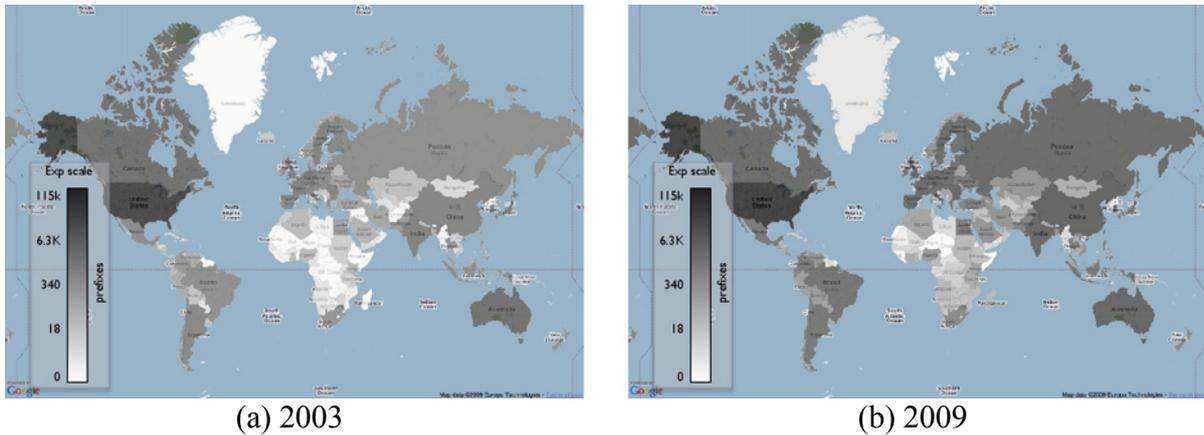


Fig. 10. Geographical distribution of the number of announced prefixes. Given on an exponential color scale, from light (least) penetration to dark (greatest).

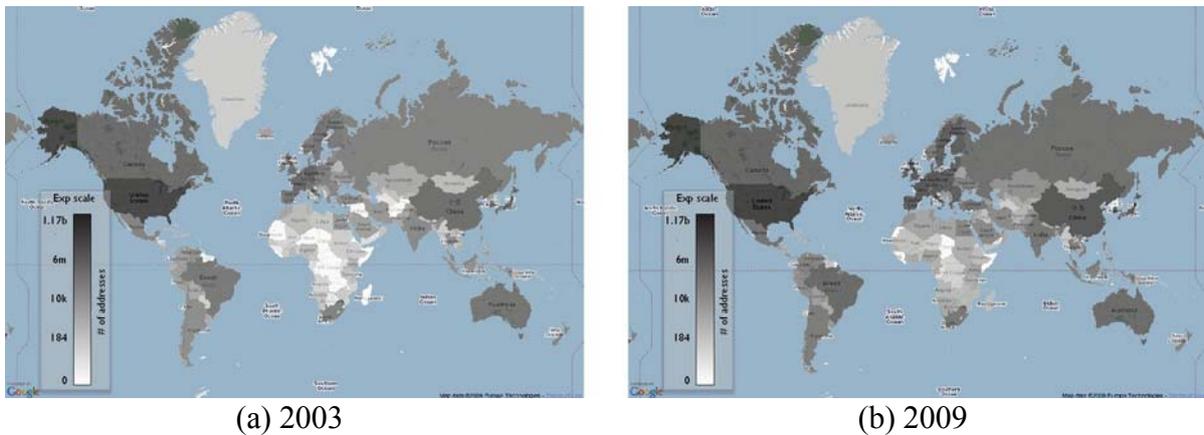


Fig. 11. Geographical distribution of the amount of announced IP space. Similar color scale as given in Fig. 10.

An extended presentation illustrating the geographical distribution of IP prefixes and IP space is available online, at <http://lasr.cs.ucla.edu/afanasyev/09-routing-map/>

References

1. Rekhter, Y., Li, T., Hares, S., et al.: RFC1771 – A border gateway protocol 4 (BGP-4). RFC (March 1995)
2. University of Oregon: Route Views Project. <http://routeviews.org/>
3. RIPE NCC: Routing Information Service (RIS). <http://www.ripe.net/ris>
4. APNIC, ARIN, RIPE NCC: IPv6 address allocation and assignment policy. ripe-466 (February 2009)
5. ARIN: Annual fee schedule. https://www.arin.net/fees/fee_schedule.html (2009)
6. Oliveira, R.V., Zhang, B., Zhang, L.: Observing the evolution of Internet as topology. In: Proc. of SIGCOMM'07 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, ACM (2007) 313–324
7. Oliveira, R., Lad, M., Zhang, B., Zhang, L.: Geographically informed inter-domain routing. In: Proc. of ICNP'2007. (2007) 103–112
8. Wang, M., Dunn, L., Mao, W., Chen, T.: Reduce IP address fragmentation through allocation. Proc. of ICCCN 2007 (August 2007) 371–376
9. Ramachandran, A., Feamster, N.: Understanding the network-level behavior of spammers. SIGCOMM Comput. Commun. Rev. 36(4) (2006) 291–302
10. Meng, X., Zhang, B., Huston, G., Lu, S.: IPv4 address allocation and the BGP routing table evolution. SIGCOMM Comput. Commun. Rev. Special Issue on Internet Vital Statistics 35(1) (January 2005) 71–80
11. Xu, Z., Meng, X., Lu, S., Zhang, L., Wittbrodt, C.J.: IPv4 Address Allocation and the Evolution of the BGP routing table. Technical Report TR-03009, UCLA Computer Science Department, Los Angeles, CA 90095, USA (March 2003)
12. Meng, X., Xu, Z., Zhang, L., Lu, S.: An analysis of BGP routing table evolution. Technical Report TR-30046, UCLA Computer Science Department, Los Angeles, CA 90095, USA (October 2003)
13. Huston, G.: IPv4 Address Report. <http://www.potaroo.net/tools/ipv4/> (2009)

ВИРТУАЛЬНЫЕ МАШИНЫ КАК ИНСТРУМЕНТАЛЬНАЯ СРЕДА САПР

Романова А.О.

Научный руководитель к.т.н., доцент Маничев В.Б
МГТУ имени Н.Э.Баумана, кафедра РК6, Москва, Россия

VIRTUAL MACHINES AS CAD/CAM/CAE/PLM/PDM ENVIRONMENT

Anastasia O. Romanova

Supervisor Ph.D., Associate professor Vladimir B. Manichev
Moscow State Technical University N.A. N.E. Bauman

Аннотация

В данной статье рассмотрены возможности использования различных систем автоматизированного проектирования (САПР) в виртуальных машинах (ВМ). В тексте приведены области применения ВМ, их преимущества и решаемые задачи, описана целесообразность применения виртуальных машин при работе с различными САПР, составлена сводная таблица оценки работы систем проектирования в ВМ.

Abstract

This article is dedicated to a problem of virtual machines (VM) used as a CAD/CAM/CAE/PDM/PLM environment. The following issues are considered: VM application; VM advantages and issues; VM application for CAD/CAM/CAE/PDM/PLM issues. A summary table of estimated performance of CAD applications running on different VMs is represented at the end of the article.

Операционная система – это программное обеспечение, занимающее особое место, как в современных прикладных компьютерных системах, так и в сложных специализированных вычислительных комплексах. Фактически, это самый первый программный продукт, который пользователи устанавливают в аппаратной среде сразу после её приобретения

На современном уровне развития технологии, пользователи имеют возможность установить на своём оборудовании несколько операционных систем, но, несмотря на это, в конкретный момент времени работать может только одна из них. Единственным способом работать с несколькими операционными системами будет постоянная перезагрузка системы, с последующим запуском одной из установленных в аппаратной среде операционных систем. В результате чего, прикладное программное обеспечение, написанное для разных операционных систем, не может одновременно выполняться на компьютере. Одним из решений этой проблемы является применение виртуальной машины.

Во многом, интерес к технологиям виртуализации обусловлен тем, что коэффициент загруженности инфраструктуры многих предприятий сегодня составляет всего 15%. Виртуализация помогает таким компаниям полностью оптимизировать существующие ресурсы и развернуть службы и приложения, способные динамически адаптироваться к меняющимся потребностям бизнеса. Она позволяет объединить информационные ресурсы в общий пул для оптимизации их использования и автоматического выполнения поступающих запросов. Объединяя в единый ресурс серверы, системы хранения данных, сетевое оборудование и другие устройства, образующие инфраструктуру предприятия, и обеспечивая совместное использование этих ресурсов разными приложениями и бизнес-процессами, виртуализация позволяет компаниям быстро реагировать на изменения внутрикорпоративной и экономической сред, начиная от колебания рабочих нагрузок, вплоть до стратегических перемен в бизнесе, на рынке и в клиентской базе.

Виртуальная машина (ВМ) — это окружение, которое представляется для «гостевой» операционной системы, как аппаратное. Однако на самом деле это программно-аппаратное окружение, которое эмулируется программным обеспечением «хостовой» системы.

Виртуальные машины могут использоваться:

- для защиты информации и ограничения возможностей процессов

- для исследования производительности программного обеспечения или новой компьютерной архитектуры
- для эмуляции различных архитектур
- с целью оптимизации использования ресурсов мэйнфреймов и прочих мощных компьютеров
- для моделирования информационных систем с клиент-серверной архитектурой на одной ЭВМ
- для упрощения управления кластерами - виртуальные машины могут просто мигрировать с одной физической машины на другую во время работы.

Так же у виртуальных машин имеется ряд преимуществ: многовариантность, независимость, изолированность и устойчивость. Виртуальные машины повышают рентабельность процесса разработки, увеличивают его скорость. Виртуальная машина увеличивает эффективность разработки.

Исходя из того, что виртуальная машина – это почти реальный персональный компьютер, то в «гостевую» ОС можно установить любое совместимое с ней программное обеспечение. Соответственно, возможна установка и работа любого САПР приложения в «гостевой» ОС.

Современные САПР можно условно разделить две категории:

| технологические | конструкторские |
|--|--|
| <ul style="list-style-type: none"> • САМ – пакет подготовки технологического процесса производства изделий. | <ul style="list-style-type: none"> • CAD - программный пакет, предназначенный для создания чертежей, и/или 3D моделей. • CAE – пакет, предназначенный для инженерных расчётов, анализа и симуляции физических процессов. • PDM/PLM – система управления жизненным циклом изделия и обеспечивающая организацию всей информацией об изделии |

Оценка целесообразности применения виртуализации при работе с конструкторскими САПР дана в таблице ниже.

Таблица 1 Целесообразность использования VM в САПР.

| САПР | Целесообразность использования VM | Комментарий |
|---------|-----------------------------------|--|
| CAD | - | Несмотря на то, что все CAD пакеты прекрасно работают внутри VM [Таблица 2], их устанавливают на «хостовой» компьютер, тем более что инженер обычно работает не более чем с 1-2 CAD одновременно. |
| CAE | + | Использование CAE пакетов внутри VM может существенно упростить и оптимизировать расчеты и расчетное время, и позволит распараллелить процессы анализа, симуляции и т.д. |
| PDM/PLM | ++ | Несомненно, именно в PDM/PLM виртуализация практически необходима. Можно привести пример импортирования и экспортирования данных из CATIA во множество систем управления информацией об изделии, таких как Nx. Особенно если использование каждой PDM/PLM не постоянное, установка на «хост» не целесообразна, а импорт/экспорт данных требуется довольно часто. |

Ниже представлена таблица, показывающая качество работы САПР приложений в инструментальной среде VM. Host OS Mac OS X 10.6.2 Leopard, Guest OS Windows XP SP3 i386, RAM 1024, 2 CPU, DirectX support enabled. Обе виртуальные машины виртуализируют свою графическую карту, где используется собственный драйвер, и полностью поддерживают DirectX 9Ex и OpenGL 2.0.

Таблица 2 Работа САПР в VM.

| | Parallels Desktop 5 | VMware Fusion 3 |
|-----------------------|---------------------|-----------------|
| • AutoCAD 2010 | + | + |
| • Inventor 2010 | + | + |
| • Micro Station v8 xm | + | + |
| • Catia r5 v17 | + | + |

Из приведенной таблицы следует, что виртуальные машины могут быть использованы для работы с любыми САПР. В целом, рассмотренные VM от компаний Parallels и VMware очень сходны между собой в отношении возможностей эмуляции и не сильно различаются как версиями поддерживаемого DirectX, так и ресурсоемкостью. Выбор конкретной VM зависит от приоритетов пользователя и поставленных задач.

Литература

1. Гульятев А. Книга Виртуальные машины - несколько компьютеров в одном. — СПб.: Питер, 2006. — С. 224.
2. <http://ru.wikipedia.org/wiki/OpenGL>
3. <http://www.apple-iphone.run/mac.html>
4. <http://www.goggle.com/trends>
5. http://en.wikipedia.org/wiki/Comparison_of_VMware_Fusion_and_Parallels_Desktop
6. <http://www.parallels.com/en/products/desktop/features/3d/>
7. <http://www.vmware.com/products/fusion/features.html>
8. <http://www.parallels.com/en/products/desktop/sr/>
9. http://www.vmware.com/products/fusion/system_requirements.html

МНОГОКРИТЕРИАЛЬНАЯ ОПТИМИЗАЦИЯ НА ОСНОВЕ НЕЧЕТКОЙ АППРОКСИМАЦИИ ФУНКЦИИ ПРЕДПОЧТЕНИЙ ЛИЦА, ПРИНИМАЮЩЕГО РЕШЕНИЯ

Д.А. Моор, Д.Т. Мухлисуллина

д.ф.-м. н., профессор Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра Ркб, Москва, Российская Федерация

Moor D.A., Mukhlisullina D. T.

Professor, Karpenko A.P.

Bauman Moscow State Technical University, Moscow, Russian Federation

Аннотация

Рассматривается прямой адаптивный метод многокритериальной оптимизации на основе аппроксимации функции предпочтения лица, принимающего решение, с помощью аппарата нечеткой логики. Исследуется эффективность метода при решении 2-х и 3-х критериальных тестовых задач с использованием различных сверток критериев оптимальности.

Abstract

The paper considers the direct adaptive method for multi-criteria optimization based on approximation of a decision maker preferences function by using the fuzzy logic apparatus. Research results of the method effectiveness for solving 2-criterial and 3-criterial tests problems with different criteria convolutions are presented.

1. Постановка задачи

Пусть X - вектор варьируемых параметров задачи. Множеством допустимых значений вектора X является ограниченное и замкнутое множество D_X . Лицо, принимающее решение (ЛПР), стремится минимизировать на множестве D_X каждый из частных критериев оптимальности $\phi_1(X), \phi_2(X), \dots, \phi_m(X)$, что *условно* записывается в виде

$$\min \Phi(X) = \Phi(X^*), X \in D_X, \quad (1)$$

где X^* - искомое решение задачи многокритериальной оптимизации (МКО-задачи). Предполагается, что частные критерии оптимальности тем или иным способом нормализованы.

Обозначим $\varphi(X, \Lambda)$ операцию свертки частных критериев оптимальности, где $\Lambda \in D_\Lambda \subset R^m$ - вектор весовых множителей; $D_\Lambda = \{\lambda_i | \lambda_i \geq 0, \sum_i \lambda_i = 1, i \in [1:m]\}$ - множество допустимых значений этого вектора. Для любого $\Lambda \in D_\Lambda$ метод скалярной свертки сводит решение задачи (1) к решению однокритериальной задачи глобальной условной оптимизации (ОКО-задачи)

$$\min \varphi(X, \Lambda) = \varphi(X^*, \Lambda), X \in D_X. \quad (2)$$

Условие (2) позволяет полагать [1], что функция предпочтений ЛПР определена не на множестве D_X , а на множестве D_Λ : $\psi: \Lambda \rightarrow R$. В результате МКО-задача сводится к задаче выбора вектора $\Lambda^* \in D_\Lambda$ такого, что $\max_{\Lambda \in D_\Lambda} \psi(\Lambda) = \psi(\Lambda^*)$, $\Lambda \in D_\Lambda$. При этом вместо функции $\psi(\Lambda)$ используется дискретная функция $\psi(\Lambda)$ [2], так что вместо указанной задачи решается задача

$$\max \psi(\Lambda) = \psi(\Lambda^*) = \psi^*, \Lambda \in D_\Lambda. \quad (3)$$

2. Метод решения задачи

Общая схема рассматриваемого прямого адаптивного метода решения МКО-задачи является итерационной и состоит из перечисленных ниже следующих основных этапов [3].

Этап «разгона» метода. МКО-система некоторым образом (например, случайно) последовательно генерирует k векторов $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ и для каждого из этих векторов Λ_i выполняет следующие действия. 1) Решает соответствующую ОКО-задачу. 2) Предъявляет ЛПР найденное решение X_i^* , а также соответствующие значения всех частных критериев оптимальности. 3) ЛПР оценивает эти данные и вводит в МКО-систему соответствующее значение своей функции предпочтений $\psi(\Lambda_i)$.

Первый этап. На основе всех имеющихся в МКО-системе значений $\Lambda_1, \Lambda_2, \dots, \Lambda_k$ вектора Λ и соответствующих оценок функции предпочтений $\psi(\Lambda_1), \psi(\Lambda_2), \dots, \psi(\Lambda_k)$ МКО-система выполняет следующие действия. 1) Строит функцию $\tilde{\psi}_1(\Lambda)$, аппроксимирующую функцию $\psi(\Lambda)$ в окрестности точек $\Lambda_1, \Lambda_2, \dots, \Lambda_k$. 2) Отыскивает максимум функции $\tilde{\psi}_1(\Lambda)$ – решает ОКО-задачу $\max \tilde{\psi}_1(\Lambda) = \tilde{\psi}_1(\Lambda^*) \quad \Lambda \in D_\Lambda$. 3) С найденным вектором Λ_1^* решает ОКО-задачу вида (2) – находит вектор параметров и соответствующие значения частных критериев оптимальности, а затем предъявляет их ЛПР. ЛПР оценивает указанные данные и вводит в систему соответствующее значение своей функции предпочтений $\psi(\Lambda_1^*)$.

Второй этап. На основе всех имеющихся в системе значений $\Lambda_1, \Lambda_2, \dots, \Lambda_k, \Lambda_1^*$ вектора Λ и соответствующих оценок функции предпочтений $\psi(\Lambda_1), \psi(\Lambda_2), \dots, \psi(\Lambda_k), \psi(\Lambda_1^*)$ МКО-система выполняет аппроксимацию функции $\psi(\Lambda)$ в окрестности точек $\Lambda_1, \Lambda_2, \dots, \Lambda_k, \Lambda_1^*$ – строит функцию $\tilde{\psi}_2(\Lambda)$ и т.д. по схеме первого этапа до тех пор, пока ЛПР не примет решение о прекращении вычислений.

3. Алгоритм и программная реализация метода

Используем в качестве свертки (2) аддитивную свертку

$$\varphi(X, \Lambda) = \sum_i \lambda_i \varphi_i(X), \quad i \in [1:m].$$

Входами системы нечеткого вывода являются значения весов частных критериев оптимальности – нечеткие термы $\lambda_{i,j,k}$, $i, j=1, 2, \dots, k \in [1:m]$. Выходной переменной системы нечеткого вывода является лингвистическая переменная ψ , ядро которой $\tilde{\psi}$ принимает значения $1, 2, \dots, 9$ [4].

Взаимосвязь между входными и выходными переменными описывается нечеткими правилами вида:

ЕСЛИ <значения входных переменных> ТО <значение выходной переменной> (f_i).

Здесь $f_i \in [0,1]$ – коэффициенты определенности (веса нечетких правил).

Используется схема нечеткого вывода Мамдани, который выполняется за два шага.

Шаг 1. Положим, что выполнено N экспериментов по определению значений лингвистической переменной ψ . Пусть в n_1 этих экспериментов переменная ψ приняла значение ψ_1 , в n_2 экспериментах – значение ψ_2 и т.д. до n_9 и ψ_9 . Соответствующие входные векторы Λ обозначим $\Lambda_{i,j} = (\lambda_{i,j,1}, \lambda_{i,j,2}, \dots, \lambda_{i,j,m}) \in \mathcal{D}_\Lambda$, где $i \in [1:9]$, $j \in [1:n_i]$.

Матрицу знаний $\{\lambda_{i,j,k}, i \in [1:9], j \in [1:n_i], k \in [1:m]\}$ можно представить в виде

$$\bigcup_{j=1}^{n_i} \left[\bigcap_{k=1}^m (\lambda_j = \lambda_{i,j,k}) \right] \rightarrow \psi_i, \quad i \in [1,9]$$

Шаг 2. Тонкая настройка модели $\psi(\Lambda)$ (параметрическая идентификация модели) осуществляется путем подбора следующих параметров: полуширина a функций принадлежности входных переменных $\mu_{i,j,k}^{\Lambda}$; полуширина b функций принадлежности выходных переменных $\mu_{i,j,k}^{\psi}$; величина δ , определяющая закон изменения весовых

множителей правил. Область допустимых значений указанных параметров представляет собой параллелепипед D_p , а критерий оптимальности имеет вид

$$E(a, b, \delta) = \sqrt{\sum (\psi(\Lambda_i) - \bar{\psi}(\Lambda_i))^2}, \quad i \in [1; N],$$

где $\bar{\psi}(\Lambda_i)$ - результат нечеткого логического вывода Мамдани в точке Λ_i .

Таким образом, формально задача параметрической идентификации формулируется в виде следующей задачи глобальной многомерной условной оптимизации [3]

$$\min E(a, b, \delta) = E(a^*, b^*, \delta^*) \quad (a, b, \delta) \in D_p. \quad (4)$$

4. Исследование эффективности метода

Исследование выполнено для двух двумерных двухкритериальных задач и одной двумерной трехкритериальной тестовых МКО-задач:

- двухкритериальная задача 1, имеющая выпуклый фронт Парето,
 $\phi_1(X) = (x_1 - 5)^2 + x_2^2 + 10$; $\phi_2(X) = x_1^2 + (x_2 - 5)^2 + 20$; $D_x = \{X \mid 0 \leq x_i \leq 5, i = 1, 2\}$;
- двухкритериальная задача 2 (невыпуклый многосвязный фронт Парето)

$$\phi_1(X) = -x_1; \quad \phi_2(X) = -f(X) \left(1 - \sqrt{\frac{x_1}{f(X)}} - \frac{x_1}{f(X)} \sin(10\pi x_1) \right);$$

$$f(X) = 1 + 9x_2; \quad D_x = \{X \mid 0 \leq x_i \leq 1, i = 1, 2\};$$

- трехкритериальная задача 3 (выпуклый фронт Парето)

$$\phi_1(X) = (x_1 - 5)^2 + x_2^2 + x_3^2 + 10; \quad \phi_2(X) = x_1^2 + (x_2 - 5)^2 + x_3^2 + 20;$$

$$\phi_3(X) = x_1^2 + x_2^2 + (x_3 - 5)^2 + 5; \quad D_x = \{X \mid 0 \leq x_i \leq 5, i = 1, 2, 3\}.$$

Исследование выполнено для $k = 3, 5, 7$ «разгонных» значений вектора Λ .

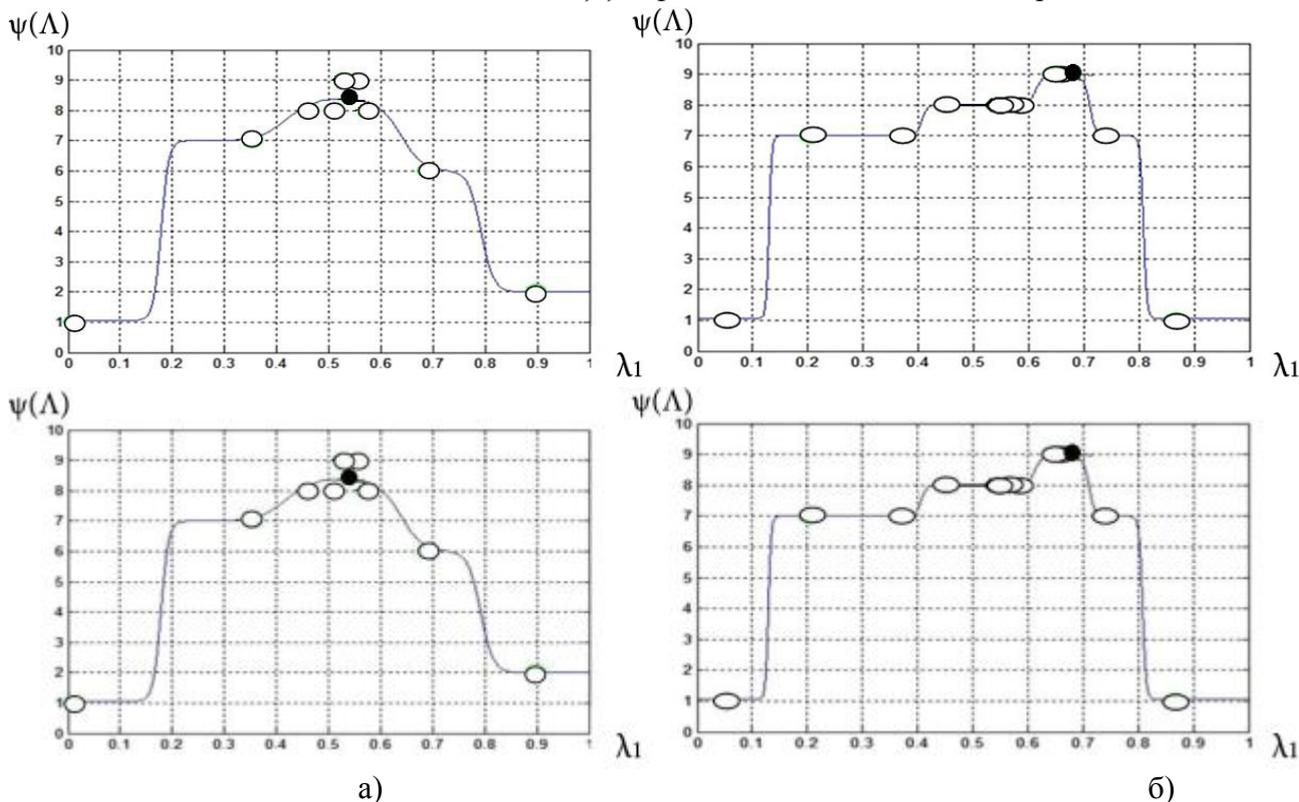


Рис. 1. Аппроксимация функция предпочтений ЛПР:

а) МКО-задача 1; $k = 3$; 12-я итерация; б) МКО-задача 2; $k = 3$; 14-я итерация

При $k = 3$ «разгонных» точка решение МКО-задачи 1 потребовало 12 итераций. Вид функции предпочтений ЛПР $\bar{\psi}(\Lambda_i)$ после 12 итераций представляет рис. 1а). Здесь и далее в число итераций включаются и «разгонные» итерации; светлые кружки соответствуют

значениям функции предпочтений ЛПР на промежуточных итерациях, темный кружок – на последней итерации. При $k=5$ «разгонных» точек решение задачи потребовало 16 итераций, при $k=7$ - 11 итераций.

Для МКО-задачи 2 при $k=3$ вид функции $\psi(\lambda_1)$ после 14 итераций представлен на рис. 2б). При $k=5$ «разгонных» точек решение задачи потребовало 14 итераций, при $k=7$ - 14 итераций.

Вид линий уровня функции $\psi(\lambda_1)$ после 17 итераций ($k=3$) решения МКО-задачи 3 показывает рис. 2. При $k=5$ «разгонных» точках решение задачи потребовало 11 итераций, при $k=7$ - 16 итераций.

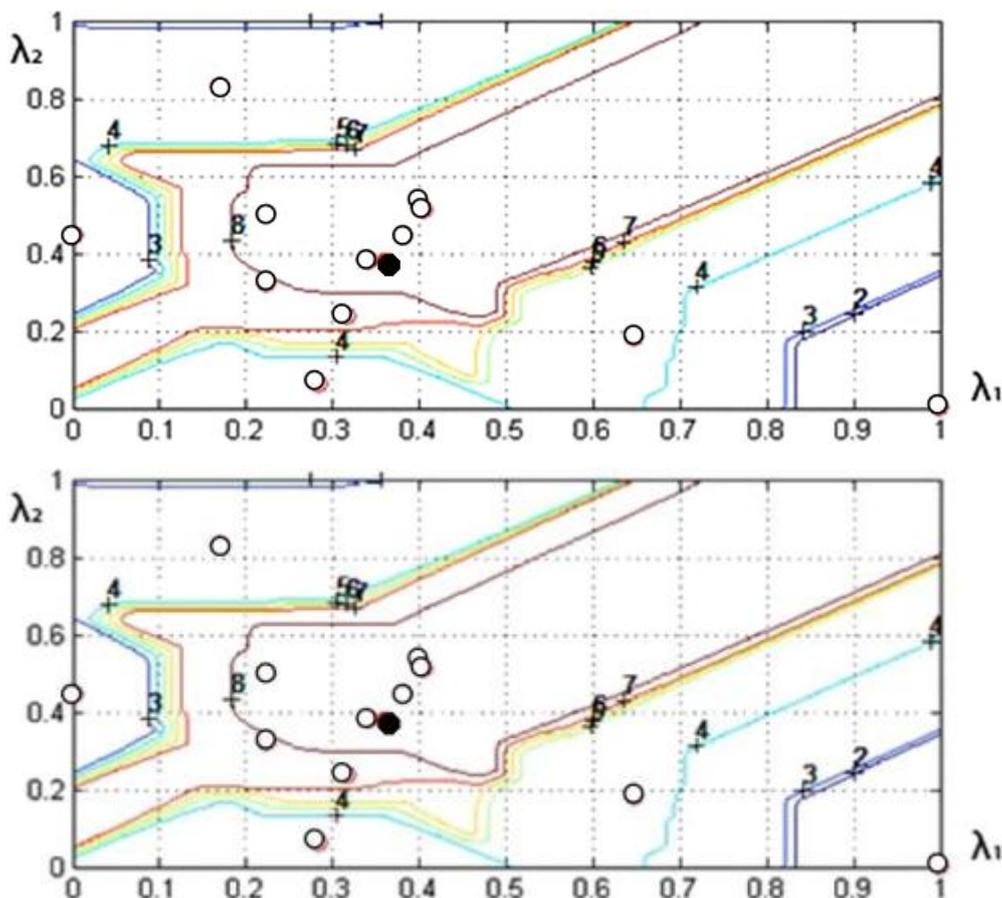


Рис. 2. Линии уровня функции, аппроксимирующей функцию предпочтений ЛПР: МКО-задача 3; $k=3$; 17-я итерация

К недостаткам метода можно отнести то, что с каждой следующей итерацией вычислительная сложность пути Фаззификация \rightarrow Агрегирование \rightarrow Активизация \rightarrow Аккумуляция \rightarrow Дефаззификация повышается, поскольку увеличивается количество входных правил нечеткой системы.

Также было проведено исследование эффективности использования сверток критериев оптимальности, отличных от линейной аддитивной свертки:

- 1) свертка Гермейера $\varphi_G(X, \lambda) = \min \phi_i(X) / \lambda_i, i \in [1 : m]$;
- 2) свертка на основе отклонения от идеальной точки $\varphi_\rho(X, X^{**}) = \rho(X, X^{**})$, где X^{**} - идеальная точка, ρ - некоторая метрика.

Заметим, что в работе в качестве метрики ρ использовалась евклидова метрика, хотя можно использовать и другие метрики, например, чебышевскую метрику $\rho(X, X^{**}, \Lambda) = \max \lambda_i |x_i - x_i^{**}|, i \in [1 : m]$.

Результаты экспериментов подтвердили, что не все точки множества Парето могут быть получены с помощью линейной аддитивной свертки (например, если множество Парето не выпукло или несвязно). Напротив, свертка Гермейера позволяет находить все точки множества Парето, включая слабо эффективные точки. Если ЛПР интересуют только точки, принадлежащие множеству Парето, это свойство свертки Гермейера следует считать ее недостатком. Недостатком свертки на основе идеальной точки является произвол в выборе метрики.

Литература

1. Лотов, А.В. Введение в экономико-математическое моделирование / А.В.Лотов.– М.: Наука, 1984.- 392 с.
2. Карпенко, А.П. Один класс прямых адаптивных методов многокритериальной оптимизации / А.П. Карпенко, В.Г. Федорук // Информационные технологии.- 2009.- №5.- С.24-30.
3. Карпенко, А.П. Нейросетевая аппроксимация функции предпочтений лица, принимающего решения, в задаче многокритериальной оптимизации / А.П.Карпенко, Д.Т. Мухлисуллина, В.А. Овчинников // Информационные технологии.- 2010.- (в печати).
4. Ротштейн, А.П. Интеллектуальные технологии идентификации: нечеткая логика, генетические алгоритмы, нейронные сети / А.П. Ротштейн.- Винница: УНИВЕРСУМ-Винница, 1999.- 320 с.

МЕТА-ОПТИМИЗАЦИЯ В ПРОЦЕССЕ НОРМАЛЬНОГО ФУНКЦИОНИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Свианадзе Зураб

д. ф.-м. н., профессор Карпенко А. П.

МГТУ им. Баумана, кафедра РК6, Москва, Российская Федерация

META-OPTIMIZATION DURING NORMAL USING OF PROGRAM

Svianadze Zurab

Professor A.P. Karpenko

Bauman Moscow State Technical University, RK6 department, Moscow, Russian Federation

Аннотация

Выбор свободных параметров алгоритмов оптимизации является одной из важнейших задач при построении производительных приложений. В данной работе рассматриваются основные подходы к выбору параметров алгоритмов, а также предлагается метод выбора свободных параметров, основанный на сборе информации о работе алгоритма в процессе его нормальной эксплуатации и анализе этой информации с помощью генетического алгоритма.

Abstract

Parameter selection in optimization algorithms is one of the most important tasks in building an efficient application. In this work discussed a classification of various methods for parameter selection. Also proposed a method, which is based on the information collecting while the algorithm is normally used, and analyzing this information with genetic algorithm.

Введение

Одним из основных этапов в проектировании поисковых алгоритмов оптимизации является выбор компонентов, из которых должен состоять алгоритм. К примеру, в эволюционных алгоритмах оптимизации необходимо выбрать операторы скрещивание и мутация, механизмы выбора родителей и потомков, начальную популяцию [1]. Каждый из этих компонентов может иметь свободные параметры, например, в эволюционных алгоритмах – это вероятность мутации, размер турнира при селекции, размер популяции и др. Значения этих параметров в значительной степени определяют эффективность алгоритма. Проблему выбора оптимальных параметров алгоритмов оптимизации выделяют в отдельный раздел – мета-оптимизация.

1 Классификация подходов к выбору параметров алгоритмов

Выбор хороших с точки зрения производительности значений параметров является трудоемкой задачей. Поэтому типичный подход к определению параметров алгоритма – выбор тех значений параметров, которые обеспечивают, в целом, хорошую производительность на широком классе задач. Такой подход называется *настройкой параметров* [1]. В этом случае создают несколько *стратегий алгоритма* (наборов параметров алгоритма) и эти стратегии тестируют на большом количестве тестовых функций. Результаты тестирования сравнивают и выбирают стратегию, показавшую наилучшую производительность. Полученный набор параметров становится рекомендуемым для исследуемого алгоритма. Например, стратегия, полученная в работе [2] при исследовании алгоритма Нелдера-Мида, принята в качестве «умолчательной» стратегии в некоторых библиотеках оптимизации (библиотека NAG). В других библиотеках значения этих параметров даже не являются настраиваемыми (библиотека системы MATLAB).

Рассмотренный подход позволяет достаточно просто выбрать стратегию алгоритма. Однако следует учитывать, что полученные таким образом параметры алгоритма не являются оптимальными для каждой конкретной решаемой задачи и не дают максимальной производительности алгоритма. Кроме того, проблемой данного подхода является

экспоненциальный рост вычислительных затрат при увеличении количества настраиваемых параметров алгоритма. Так, уже при 4 свободных параметрах и всего 5 возможных значениях каждого из них, необходимо протестировать $5^4 = 625$ стратегий. Если тестирование соответствующего алгоритма производится на 100 задачах, то это потребует 62500 запусков алгоритма.

Существуют алгоритмы, в которых некоторые параметры желательно сделать динамическими. Примерами таких параметров являются вероятность мутации в генетических алгоритмах, скорость обучения в алгоритме обратного распространения ошибки. Значения этих параметров должны уменьшаться по мере работы алгоритмов. Наилучшим вариантом было бы иметь функцию, определяющую значения таких параметров в зависимости от числа выполненных итераций и, быть может, некоторой другой информации о текущих результатах работы алгоритма. Однако ввиду большой сложности поиска такой функции, обычно ограничиваются введением некоторого априорного правила, по которому параметры изменяются в процессе работы алгоритма. Примером такого правила является охлаждающее правило в алгоритме имитации отжига.

Для эволюционных алгоритмов характерно наличие большого числа настраиваемых параметров. Поэтому для разработчиков этих алгоритмов задача выбора правильной стратегии алгоритма стоит наиболее остро. Именно в этом классе алгоритмов оптимизации получил развитие другой подход к выбору свободных параметров – *управление параметрами*. Данный подход подразумевает использование информации о результатах работы алгоритма. При управлении параметрами пытаются решить проблему выбора оптимальной стратегии для конкретной задачи, а не для класса задач. Поэтому при успешном решении задачи управления параметрами можно добиться значительного прироста производительности алгоритма при решении данной конкретной задачи. Отметим, что при управлении параметрами автоматически решается проблема поиска динамических параметров, поскольку в этом случае все параметры являются динамическими и изменяются в процессе работы алгоритма.

Управление параметрами в эволюционных алгоритмах можно разделить на два вида.

а) *Адаптивное управление параметрами*. Здесь механизм выбора параметров выделен явно и является надстройкой над алгоритмом. На основе данных о работе алгоритма определяются направления изменения параметров и величины изменения значений этих параметров.

б) *Самоадаптивное управление параметрами*. В этом случае эволюционный алгоритм оптимизирует себя сам. Свободные параметры алгоритма расширяют пространство поиска и вносят свой вклад в целевую функцию. Например, в генетических алгоритмах свободные параметры зашифровываются в хромосомы наравне с другими параметрами и подвергаются мутациям и рекомбинациям. Механизм изменения параметров в этом случае неотделим от самого алгоритма.

В работе [1] приводятся результаты применения различных видов управления параметрами в эволюционных алгоритмах.

2 Управление параметрами алгоритмов оптимизации

Управление параметрами применимо не только к эволюционным алгоритмам, но и к другим алгоритмам оптимизации.

Рассмотрим класс T задач глобальной безусловной оптимизации

$$\min_{X \in R} \phi(X) = \phi(X^*) = \phi^*.$$

Известно, что эффективность решения задачи $t \in T$ любым из методов безусловной глобальной оптимизации в значительной мере зависит от класса функций, которому принадлежит функция $\phi(X)$, а также от ряда других характеристик этой функции:

- константа Липшица;
- максимальное значение градиента (если функция $\phi(X)$ дифференцируема);

- количество локальных минимумов;
- вычислительная сложность.

Вектор свободных параметров алгоритма обозначим Q . Например, для алгоритма метода Нелдера-Мида такими параметрами являются параметры α , β , γ – коэффициенты растяжения, сжатия и редукции многогранника соответственно.

Предполагается заданным множество D_Q допустимых значений компонентов вектора параметров Q :

$$D_Q = \{Q \mid q_i \in [q_i^-, q_i^+], i = 1, 2, \dots\}.$$

Все возможные допустимые значения вектора параметров Q определяют множество алгоритмов

$$A = \{a \mid Q \in D_Q\}.$$

Будем различать априорную и апостериорную информацию о характеристиках задачи. Априорная информация задается пользователем до решения задачи, апостериорная информация получается в процессе решения задачи $t \in T$. Необходимость выделения априорной и апостериорной информации обусловлены тем, что не все характеристики задачи могут быть известны заранее, а известные характеристики могут быть приблизительными. С другой стороны характеристики задачи могут быть оценены в процессе решения задачи.

Введем в рассмотрение два вектора признаков задачи t : $In(t)$ – вектор признаков, определяющих априорную информацию о задаче; $Out(t)$ – аналогичный вектор, определяющий апостериорную информацию.

Для задачи $t \in T$ оптимальный вектор Q^* значений параметров алгоритма $a \in A$ зависят от значений априорных и апостериорных признаков этой задачи, т.е.

$$Q^* = Q^*(t) = W(In(t), Out(t)),$$

где W – некоторая неизвестная функция. Алгоритм $a^*(Q^*) \in A$, т.е. алгоритм, который использует значения параметров Q^* , будем называть оптимальным алгоритмом решения задачи $t \in T$.

Предлагается 1) управлять параметрами алгоритма оптимизации A , применяемого для решения задачи безусловной глобальной оптимизации $t \in T$, на основе априорной и апостериорной информации о характеристиках задачи; 2) отыскивать оптимальную стратегию для алгоритма в процессе нормальной эксплуатации программного обеспечения, реализующего данный алгоритм.

Для реализации данной идеи предложено использовать клиент-серверную модель. При этом пользователь, запускающий алгоритм для решения задачи оптимизации, формирует вектор априорной информации о задаче и отправляет его на сервер. Сервер в ответ отправляет клиенту вектор параметров задачи. Клиент решает задачу с использованием полученных параметров. В процессе решения задачи происходит обмен информацией клиента с сервером, в процессе которого клиент предоставляет серверу апостериорную информацию о решаемой задаче, получая в ответ уточненные значения параметров алгоритма.

По информации о характеристиках задачи на сервере определяется класс, к которому относится данная задача. Для этого решается соответствующая задача кластеризации. Оптимальный набор значений параметров алгоритма определяется для каждого класса задач. Задача кластеризации решается с помощью нейронных сетей типа самоорганизующихся карт Кохонена.

Для выбора значений параметров алгоритма на сервере используется генетический алгоритм. Задача выбора параметров плохо формализована, слабо изучена, обладает высокой сложностью. Известно, что генетические алгоритмы в этом случае работают эффективнее других методов. Дополнительным плюсом является то, что генетические алгоритмы хорошо исследуют пространства поиска высокой размерности, что делает возможным управление

большим количеством настраиваемых параметров алгоритма. В качестве критерия оптимальности взята линейная свёртка из оценки вероятности локализации минимума функции, математического ожидания и дисперсии количества испытаний, необходимых для решения задачи.

Количество испытаний (вычислений значений функции $\phi(X)$), необходимых для решения задачи, характеризует время, которое требуется для решения задачи данным алгоритмом. Вообще говоря, это время складывается из двух основных компонентов – суммарного времени на испытания и времени вычисления новых компонент вектора X на основе его предшествующих значений. В практически важных задачах время на испытания значительно превышает время расчета компонент вектора X из-за высокой вычислительной сложности функции $\phi(X)$. Это обстоятельство позволяет использовать в качестве критерия количество испытаний, пренебрегая временем вычисления новых компонент вектора X .

Проведены тестовые испытания предложенного подхода. В качестве алгоритма глобальной безусловной оптимизации использована комбинация алгоритма метода Нелдера-Мида, как достаточно распространенного алгоритма локальной безусловной оптимизации, и метода мултистарта. Испытания проводились на тестовых функциях Розенброка и Химмельблау. В результате эволюции параметров алгоритма Нелдера-Мида с помощью генетического алгоритма минимизация тестовых функций выполнялась за число испытаний на 15-20% меньшее, чем с рекомендуемыми параметрами этого алгоритма.

Полученные результаты показали, что данный подход может увеличить эффективность применения алгоритмов оптимизации. В реальных задачах САПР, которые обладают высокой вычислительной сложностью и требуют больших вычислительных затрат на проведение каждого испытания, повышение производительности алгоритма может дать существенный выигрыш во времени проектирования, несмотря на дополнительные коммуникационные расходы, связанные с обменом сообщениями между клиентом и сервером. Поэтому работы по развитию представленной модели мета-оптимизации продолжаются.

Литература

1. Z. Michalewicz, A. E. Eiben, R. Hinterding // Evolutionary Optimization.- New York, Springer, 2003.- V. 48.- pp. 279-306.
2. J. A. Nelder, R. Mead. A simplex method for function minimization // Computer Journal, 1965, №7.- :pp. 308-313.

ПРИМЕНЕНИЕ ГЕНЕТИЧЕСКОГО МЕТОДА КОМБИНИРОВАНИЯ ЭВРИСТИК ДЛЯ ПРОЕКТИРОВАНИЯ МНЕМОСХЕМЫ ДИСПЕТЧЕРСКОГО ЩИТА ЭЛЕКТРИЧЕСКОЙ СЕТИ

Кузьмин А.А.

Научный руководитель: д.т.н., профессор, заведующий кафедрой РК-6 И.П. Норенков
Московский Государственный Технический Университет им. Н.Э.Баумана, Москва, РФ

APPLICATION OF HEURISTICS COMBINATION METHOD FOR ELECTRIC SYMBOLIC CIRCUIT DESIGN

Kuzmin A.A.

Scientific adviser: Ph.D., professor, head of department RK-6 I.P. Norenkov
Moscow State Technical University named after Bauman, Moscow, Russian Federation

Аннотация

В данной работе рассмотрены подходы к решению задачи автоматизированного проектирования мнемосхемы диспетчерского щита электрической сети. Формализован критерий наглядности мнемосхемы диспетчерского щита электрической сети. Предложен алгоритм решения указанной задачи, основанный на методе комбинирования эвристик.

Abstract

In this paper the approaches of solving of the problem of design of electronetwork symbolic circuit are considered. The criterion of obviousness of electronetwork symbolic circuit is formalized. The algorithm of solving of the mentioned problem based on Heuristics Combination Method is proposed.

Введение

В данной работе рассмотрены вопросы оптимизации процессов оперативно-диспетчерского управления электрическими сетями, а именно, автоматизация проектирования мнемосхемы диспетчерского щита (ДЩ). ДЩ является основным инструментом диспетчера, позволяющим оперативно принимать решения, в том числе, в случае аварии. Целью решения рассматриваемой задачи является оптимальное размещения элементов мнемосхемы и связей между ними в монтажном пространстве ДЩ с точки зрения критерия «наглядности». «Наглядность» в данном случае характеризуется эффективностью работы диспетчера с ДЩ.

Наиболее близкими к указанной задаче являются задачи размещения и трассировки в микроэлектронике [3 - 4]. Следует отметить, что ввиду большой размерности рассматриваемой задачи решение ее полным перебором не представляется возможным. Кроме того, ввиду сложности формализации указанного выше критерия «наглядности» и связанной с этим специфики задачи, непосредственное использование последовательного и итерационного алгоритмов размещения для ее решения не представляется возможным. В данной работе предлагается использовать алгоритм, основанный на методе комбинирования эвристик с использованием последовательного алгоритма размещения.

Постановка задачи

Исходные данные

1. Мнемосхема электрической сети

-множество конструктивных элементов: $\mathcal{E} = \{e_i \mid i = \overline{1, n}\}, n \in Z;$

-множество соединяющих их цепей: $C = \{c_r \mid r = \overline{1, k}\}, k \in Z;$

-структура электрической сети представляется неориентированным графом $G(X, U): \mathcal{E} \leftrightarrow X, C \leftrightarrow U;$

2. Параметры элемента

- тип (РП, ТП, РТП) - определяет габариты, количество возможных присоединений и т.д.;
 - географические координаты;
3. Параметры монтажного пространства
- модель монтажного пространства – неориентированный граф G_r (граф решетки);
 - габариты;
 - множество фиксированных позиций для установки элементов
- $$T = \{t_j \mid j = \overline{1, m}\}, m > n, m \in Z.$$

Ограничения

1. Пространство размещения элементов ограничено размерами щита;
2. Размещенные объекты не должны пересекаться;
3. Элементы в цепочке размещаются с заданным шагом;
4. Минимальное расстояние между цепочками задано.

Частные критерии оптимальности

Рассмотрим критерии, формализующие понятие «наглядности» и являющиеся специфическими для данной задачи.

1. Топологический критерий

Данный критерий характеризует качество размещения подстанций с точки зрения их объединения в цепочки подстанций (КТ).

Цепочка подстанций – размещение подстанций типа ТП между двумя подстанциями типа РП на заданном расстоянии, все подстанции типа ТП имеют одинаковую ориентацию.

Необходимо, чтобы число подстанций, объединенных в цепочки, было максимальным, т.е. необходимо максимизировать величину:

$$K_{\text{кол}} = n_{\text{цеп}} / n_{\text{ТП}}, \quad (1)$$

где $n_{\text{цеп}}$ - количество подстанций типа ТП, входящих в цепочки, $n_{\text{ТП}}$ - количество ТП

Кроме этого, количество цепочек подстанций $m_{\text{цеп}}$ должно быть минимальным, т. е. необходимо минимизировать:

$$K_{\text{цеп}} = m_{\text{цеп}} / n_{\text{ТП}}, \quad (2)$$

Объединяя критерии (1) и (2), определим топологический критерий в виде

$$\max \leftarrow K_T = \beta K_{\text{кол}} + (1 - \beta)(1 - K_{\text{цеп}})$$

где β – весовой коэффициент.

2. Географический критерий

Данный критерий характеризует степень соответствия между теоретическим размещением подстанций и существующим (K_G)

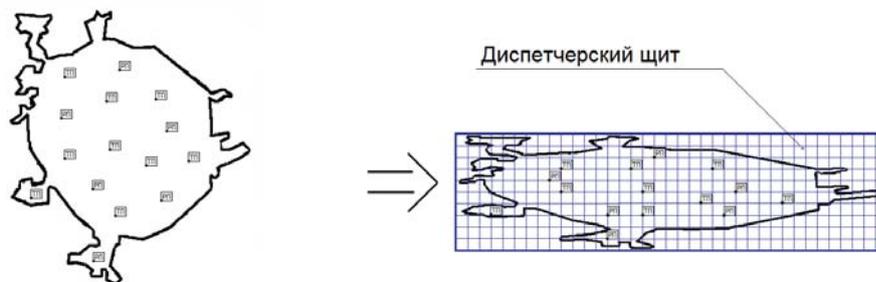


Рис.1. Географическая привязка

Данный критерий равен среднему относительному расстоянию между позицией размещения объекта в монтажном пространстве (i, j) и существующим географическим положением объекта (p, q), перенесенным на диспетчерский щит в соответствующем масштабе (см. рис. выше):

$$\min \leftarrow K_r = \frac{1}{n\sqrt{X^2 + Y^2}} \sum_{k=1}^n \sqrt{(i-p)^2 + (j-q)^2}$$

где n – количество размещаемых объектов, X, Y – размеры щита

3. Эргономический критерий (K_3)

Данный критерий характеризует, насколько просто и наглядно размещены электрические линии связи между позициями подстанций в монтажном пространстве диспетчерского щита.

Электрическая линия размещена эргономично, если она имеет минимальное количество изгибов – переходов от горизонтального направления к вертикальному и наоборот, а также минимальное количество пересечений с другими линиями.

Относительное общее количество изгибов для всех электрических линий: $K_{Изг} = m_{Изг}/XY$, где $m_{Изг}$ – общее количество изгибов для всех электрических линий

Относительное общее количество пересечений для всех линий: $K_{Пер} = m_{Пер}/XY$, где $m_{Пер}$ – общее количество пересечений для всех электрических линий.

Таким образом, эргономический критерий можно определить следующим образом:

$$\min \leftarrow K_3 = \gamma K_{изг} + (1 - \gamma) K_{пер}$$

где γ – весовой коэффициент.

Обобщенный критерий оптимальности

Данный критерий объединяет Топологический, Географический и Эргономический критерии и характеризует «наглядность» схемы:

$$\max \leftarrow K = \alpha_1 K_r + \alpha_2 (1 - K_r) + \alpha_3 (1 - K_3)$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 1$$

где $\alpha_1, \alpha_2, \alpha_3$ – весовые коэффициенты

Алгоритм решения задачи

Для решения поставленной задачи используется генетический метод комбинирования эвристик. Установка элементов в монтажном пространстве ДЩ осуществляется последовательным алгоритмом размещения. Трассировка линий связи производится преимущественно с применением волнового алгоритма.

При этом

1. гены представляют из себя эвристики, определяющие размещение очередного элемента в монтажном пространстве и проведение линий связи на текущем шаге;
2. аллели генов – номера соответствующих эвристик;
3. длина хромосомы равна количеству размещаемых элементов.

Таким образом, каждая хромосома представляет собой решение исходной задачи, которое получено путем последовательного размещения элементов в монтажном пространстве в соответствии с используемой на каждом шаге эвристикой.

Каждая эвристика состоит из трех частей: выбор подстанции, размещение подстанции и размещение линии связи. Ниже приведены эвристики, использованные для решения задачи:

| № | Обозн. | Описание |
|---|--------|--|
| 1 | A | Выбрать подстанцию с наибольшим количеством связей со <i>всеми</i> другими подстанциями |
| 2 | B | Выбрать подстанцию с наибольшим количеством связей с <i>ранее выбранными</i> подстанциями |
| 3 | C | Выбрать подстанцию, которую можно разместить максимально близко к ее географической привязке |
| 4 | D | Выбрать подстанцию с наибольшим количеством связей с последней размещенной (удлинение цепочки) |

Таблица 1. Выбор подстанции

| № | Обозн. | Описание |
|---|--------|--|
| 1 | a | Разместить подстанцию в геометрическом центре множества инцидентных подстанций |
| 2 | b | Разместить подстанцию в "центре масс" множества инцидентных подстанций |
| 3 | c | Разместить подстанцию на минимальном (оптимально – нулевом) расстоянии от ее географической привязки |
| 4 | d | Разместить подстанцию типа ТП по направляющей цепочки подстанций максимально близко к последней размещенной подстанции (удлинение цепочки) |
| 5 | e | Разместить подстанцию максимально близко к одной из инцидентных из числа размещенных (удлинение цепочки) |

Таблица 2. Размещение подстанции

Линия связи представляет собой отрезок, являющийся продолжением направляющей цепочки, в случае, когда очередной элемент дополняет одну из цепочек. В остальных случаях линия связи проводится с применением волнового алгоритма.

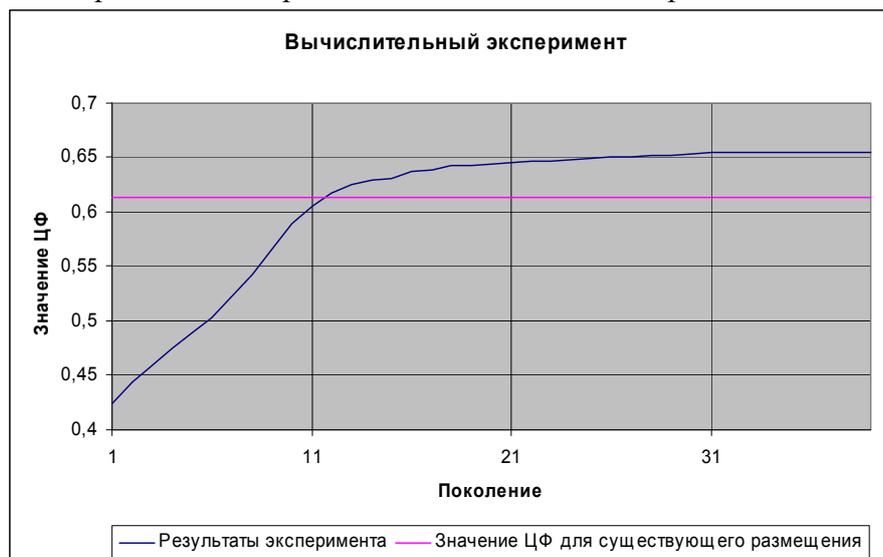


Рис.2. Вычислительный эксперимент

Литература

1. Белоусов А. И., Ткачев С. Б. Дискретная математика: Учеб. для вузов / Под ред. В. С. Зарубина, А. П. Крищенко. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2004. – 744 с.
2. Овчинников В. А. Алгоритмизация комбинаторно-оптимизационных задач при проектировании ЭВМ и систем: Учеб. для вузов. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2001. – 228 с.
3. Савельев А.Я., Овчинников В.А. Конструирование ЭВМ и систем. М.: Высшая школа, 1989. 812с.
4. Мелихов А.Н., Бернштейн Л.С., Курейчик В.М. Применение графов для проектирования дискретных устройств. М.: Наука, 1974. 303с.

ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОБНАРУЖЕНИЯ ЛИЦА ЧЕЛОВЕКА НА ИЗОБРАЖЕНИИ

Туманов А.В.

Научный руководитель: к.т.н., доцент Волосатова Тамара Михайловна
МГТУ им. Н.Э.Баумана кафедра РК6, Москва, Россия

FACE DETECTION USING GENETIC ALGORITHM

Tumanov A.V.

Scientific chief: Ph.D., senior lecturer Volosatova Tamara Mikhaylovna
MSTU n.a. Bauman PK6, Moscow, Russia

Аннотация

В работе представлен метод обнаружения и локализации лица человека на цифровых изображениях с использованием генетического алгоритма. Разработан подход к решению поставленной задачи с помощью этого алгоритма. Рассмотрена возможность применения классического генетического алгоритма. Представлен модифицированный вариант генетического алгоритма. Проведен анализ эффективности обнаружения как при использовании классического, так и модифицированного алгоритмов.

Abstract

This paper presents a method for detecting faces in images using genetic algorithm. New approach to face detection using genetic algorithm is designed. Possibility of using classic genetic algorithm is considered. Specific genetic algorithm is developed. Effectiveness of both classic and specific genetic algorithms is analyzed.

Введение

Генетические алгоритмы — это процедуры поиска, основанные на механизмах естественного отбора и наследования. В них используется эволюционный принцип выживания наиболее приспособленных особей. Эти алгоритмы успешно применяются в решении широкого спектра задач, таких как оптимизация функций, оптимизация запросов в базах данных, разнообразные задачи на графах (задача коммивояжера, раскраска), настройка и обучение искусственной нейронной сети, задачи компоновки и др. В данной работе проводится исследование возможности применения генетического алгоритма для решения задачи обнаружения лица человека на плоском изображении.

Проблема обнаружения лица и его отличительных черт на изображениях не является новой, и существует уже достаточно эффективных подходов для решения данной задачи, такие как использование нейронных сетей, цепей Маркова, различных алгоритмов цветовой сегментации и др.

В данной работе делается попытка с помощью применения генетического алгоритма и деформируемого шаблона лица человека разработать новый подход для решения проблемы устойчивой и эффективной локализации лица человека на изображении при различных условиях съемки.

Описание метода обнаружения

В данной работе применяется генетический поиск на основе деформируемого шаблона для обнаружения лица человека на изображении. Общая схема данного подхода представлена на рис. 1



Рис. 1. Общая схема предлагаемого метода

Предобработка

В качестве исходного изображения предлагается использовать цветное изображение, получаемое с любого устройства регистрации, либо из графического файла одного из распространенных форматов. Сначала изображение переводится в цветовую палитру градации серого, путем исключения из RGB палитры красной составляющей. Выбирается именно красная, потому что именно эта составляющая наиболее сильно влияет на цветовой оттенок кожи человека.

Далее осуществляется гистограммная нормализация изображения для устранения слабой контрастности изображения из-за плохих условий съемки, либо других сторонних факторов.

Генетический поиск

Идея генетических алгоритмов заимствована у живой природы и состоит в организации эволюционного процесса, конечной целью которого является получение оптимального решения в сложной комбинаторной задаче.

Задача кодируется таким образом, чтобы её решение могло быть представлено в виде вектора, также называемого хромосомой. Обычно для кодирования используется двоичная кодировка. На первом этапе генетического поиска создаётся некоторое количество начальных хромосом, называемых начальной популяцией. Они оцениваются с использованием функции приспособленности, в результате чего каждой хромосоме в популяции присваивается определённое значение приспособленности, которое определяет вероятность выживания организма, представленного данной хромосомой. После этого с использованием полученных значений приспособленности осуществляется селекция, т.е. выбор хромосом, допущенных к скрещиванию. К этим хромосомам применяются генетические операторы, создавая таким образом следующее поколение. В классическом генетическом алгоритме в качестве генетических операторов применяются операторы скрещивания и мутации. Особи следующего поколения также оцениваются, затем производится селекция, применяются генетические операторы и т.д., пока не будет достигнуто условие окончания выполнения, в качестве которого чаще всего применяют ограничение количества поколений, либо достижение функции приспособленности определённого значения для какой-либо хромосомы в поколении.

Как уже ранее отмечалось, генетический поиск в предлагаемом методе осуществляется на основе деформируемого шаблона. Шаблон называется деформируемым, так как в процессе эволюции генетического алгоритма исходный шаблон претерпевает изменения, что отражается в каждой из особей популяции на каждом поколении. Каждая хромосома содержит в себе закодированную информацию о координатах расположения шаблона относительно исходного изображения, коэффициенте масштаба и коэффициенте яркости шаблона (рис.2). Исходный шаблон представлен на рис.3

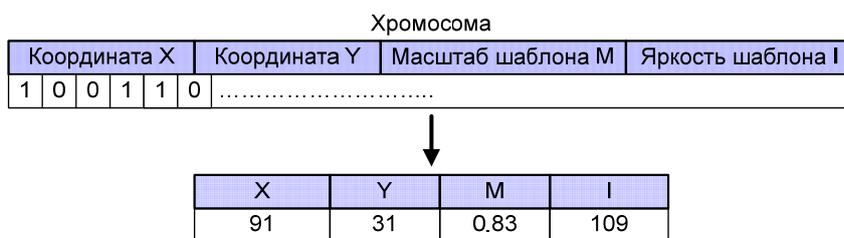


Рис. 2 Представление хромосомы для деформируемого шаблона при генетическом поиске



Рис. 3 Исходный деформируемый шаблон

Функция приспособленности

В качестве функции приспособленности (фитнес функции) предлагается использовать коэффициент взаимной корреляции, представленный далее:

$$r_{x,y} = \frac{\sum_x \sum_y \delta_I(x,y) \delta_T(x,y)}{\sqrt{\sum_x \sum_y \delta_I(x,y)^2} \sqrt{\sum_x \sum_y \delta_T(x,y)^2}},$$

где

$$\delta_I(x,y) = I_I(x,y) - \bar{I}_{I(x,y)},$$

$$\bar{I}_{I(x,y)} = \frac{1}{m \cdot n} \sum_x \sum_y I_I(x,y),$$

$$\delta_T(x,y) = I_T(x,y) - \bar{I}_{T(x,y)},$$

$$\bar{I}_{T(x,y)} = \frac{1}{m \cdot n} \sum_x \sum_y I_T(x,y),$$

$$x \in \{0, \dots, m\},$$

$$y \in \{0, \dots, n\}$$

В данной формуле I_I - означает значение интенсивности серого в исходном изображении, I_T - в шаблоне, m - размер, сравниваемых изображений по x , n - размер, сравниваемых изображений по y . Значение $r_{x,y}$ варьируется в следующем пределе $[-1;1]$ в зависимости от соответствия одного изображения другому. -1 говорит о том, что изображения полностью отличаются друг от друга, 1, то что полностью идентичны. Общая блок-схема алгоритма представлена на рис. 4.

В процессе исследований применимости генетического алгоритма для решения поставленной задачи была сделана попытка использования классического алгоритма, в котором качестве оператора селекции применялся турнирный метод, оператора скрещивания двухточечное скрещивание и стандартный оператор мутации. Исследование эффективности проводилось на базе данных лиц СИТ, состоящий из 450 цветных изображений лиц людей, полученных при различных условиях съемки, каждое из которых на одном изображении содержит одно лицо.

В качестве меры эффективности оценивался процент правильных обнаружений и скорость сходимости решения. При применении классического алгоритма процент правильных решений примерно составил 61 %, что объясняется графиками, представленными на рис. 5, на которых отражены максимальное и среднее значение фитнес функции в процессе эволюции для 2 изображений из базы данных СИТ при 100 особях в популяции. В процессе эволюции каждая из хромосом должна стремиться увеличить свою

функцию приспособленности, что должно привести к увеличению максимального значения фитнеса в популяции и приближению среднего значения к максимальному. Из графиков на рис. 5 видно, что при использовании классических алгоритмов не происходит планомерного увеличения значений функции приспособленности.

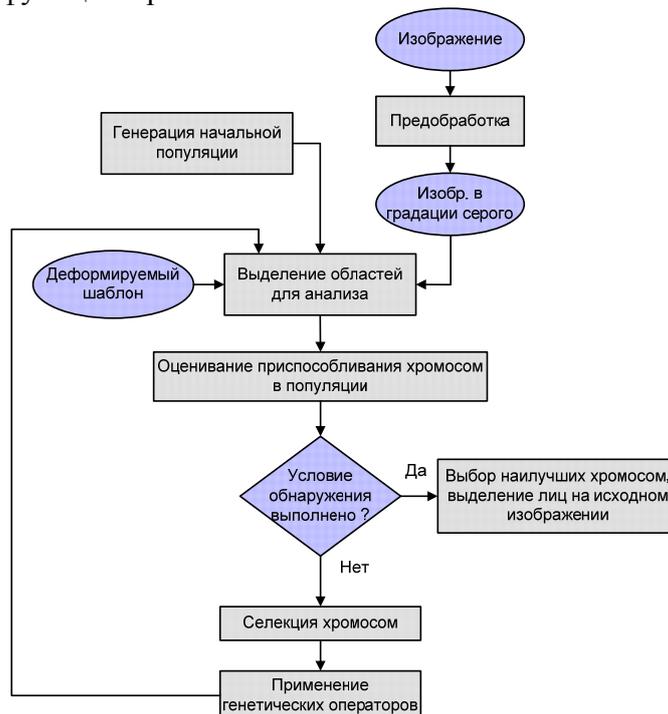


Рис. 4 Блок-схема генетического поиска

Проанализировав результаты, полученные при использовании классического генетического алгоритма, был разработан модифицированный генетический алгоритм, описание операторов которого представлено на рис. 6.



Рис. 5 Графики среднего и максимального значений фитнес функции в процессе эволюции при применении классического генетического алгоритма

В предлагаемом алгоритме не существует формального оператора селекции. Вместо этого на первом этапе отбора осуществляется сортировка всех особей в 2 группы в зависимости от значений их фитнес функции. Элитная группа может состоять из L процентов от всех хромосом в популяции. Величина L может изменяться в процессе эволюции. На этапе скрещивания каждая из элитных хромосом переходит в следующее поколение, а каждая из простых хромосом случайным образом скрещивается с применением двухточечного кроссовера с одной из элитных хромосом, в результате которого в новое поколение добавляется еще одна особь. Следует отметить, что скрещивание осуществляется по каждой из компонент хромосомы отдельно. К каждой из хромосом до попадания в новое поколение с вероятностью M может быть применен классический оператор мутации.

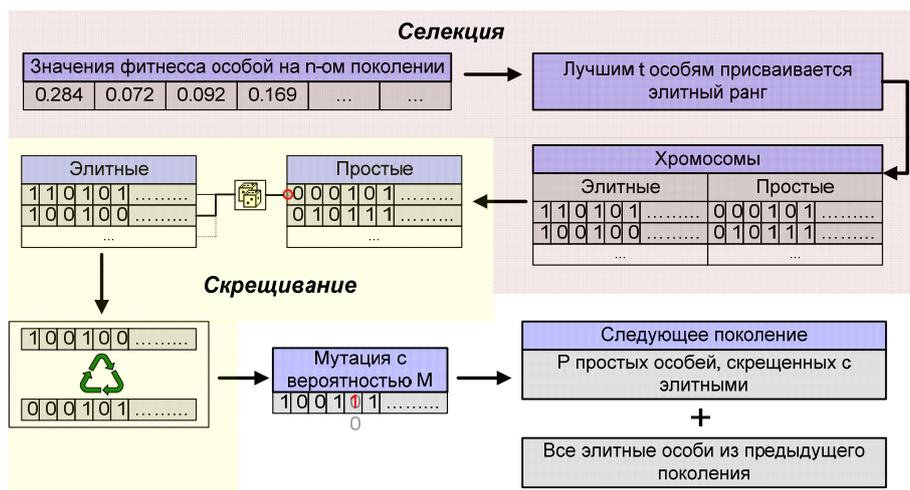


Рис. 6 Схема работы операторов предлагаемого модифицированного генетического алгоритма

Применение описанного генетического алгоритма позволило увеличить процент правильного обнаружения до 90 %. Графики среднего и максимального значения фитнеса для тех же изображений, что и при тестировании классического алгоритма, представлены на рис.7.

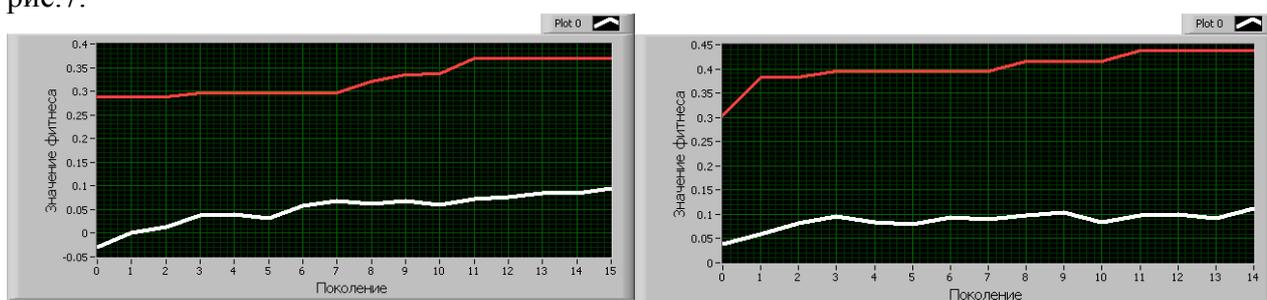


Рис. 7 Графики среднего и максимальных значений фитнес функции в процессе эволюции при применении модифицированного генетического алгоритма

Заключение

Проведенное исследование показало возможность применения генетических алгоритмов в задаче обнаружения лица человека на изображении. Решение проблемы обнаружения лица человека на изображении может быть первым этапом более масштабной задачи распознавания человека по его фотопортрету. Опыт и результаты, полученные в ходе данного исследования, будут использованы при последующей разработке и исследовании генетических алгоритмов для решения задачи распознавания человека по фотопортрету. В дальнейшем также планируется доработать предложенную методику обнаружения, добавив дополнительный компонент в хромосому деформируемого шаблона, который будет учитывать угол поворота головы человека относительно вертикали.

Литература

1. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы.– Горячая линия – Телеком, 2006. – 452 с.
2. Самаль Д.И., Старовойтов В.В. Методика автоматизированного распознавания людей по фотопортретам // Цифровая обработка изображений. - Минск:ИТК, 1999.г.
3. [Л. А. Гладков, В. В. Курейчик В. М. Курейчик](#). Генетические алгоритмы. - [ФИЗМАТЛИТ](#), 2006 г. – 320 с.

АДАПТИВНОЕ УПРАВЛЕНИЕ СЕКЦИЕЙ МАНИПУЛЯТОРА ТИПА ХОБОТ

С.М. Гоменюк

Научные руководители к.т.н, доцент В.М. Буянкин,
д. ф.-м. н., профессор А.П. Карпенко
МГТУ имени Н.Э.Баумана, кафедра РК6, Москва, Россия

ADAPTIVE CONTROL FOR A SECTION OF THE TRUNK MANIPULATOR

S. Gomenyuk

Supervisors V. Buyankin, A. Karpenko
Bauman Moscow State Technical University

Аннотация

Рассматриваются проблемы построения систем управления многосекционным манипулятором типа «хобот». Приводятся варианты построения адаптивных систем управления секцией манипулятора на базе эталонной модели, а так же на основе нейросетевого регулятора на базе инверсной модели.

Abstract

This article considers the construction problem of the control system for one section of the multiple-section trunk manipulator. The text contains the adaptive control system based on the reference model as well as on the basis of the neural network controller based on the inverse model.

Постановка задачи. В зависимости от выполняемых функций выделяется несколько типов манипуляторов типа «хобот», которые различаются типами механизмов параллельной кинематики, на основе которых построены их секции. Конструкция секции однозначно определяет количество подвижностей механизма, а, следовательно, и количество управляющих параметров [1].

Рассмотрим секцию манипулятора типа «трипод» с двумя степенями свободы, схема которого представлена на рис. 1. Здесь a, b, c - сферические шарниры, A, B, C, d - шарниры Гука; 1,2,3 – поступательные кинематические пары.

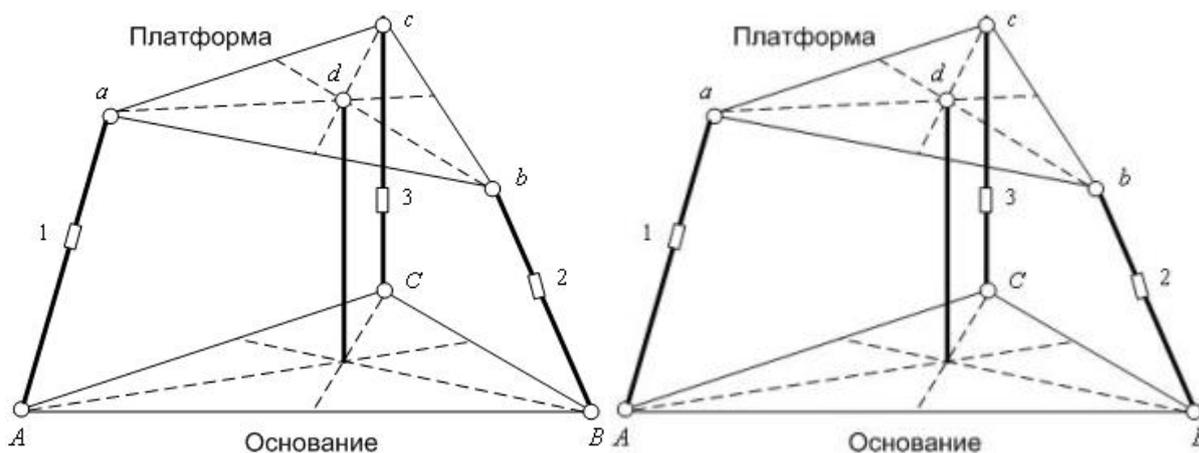


Рис. 1. Схема трипода с двумя степенями свободы

Кинематические и динамические уравнения рассматриваемого трипода получены в работе [1]. Кинематические уравнения представляют собой зависимость длин штанг трипода от требуемых углов поворота платформы. Динамические уравнения описывают закон углового движения платформы при изменении длин штанг и получены в виде системы обыкновенных дифференциальных уравнений 2-го порядка.

В качестве исполнительных устройств системы автоматического управления (САУ) триподом используются три двигателя постоянного тока. Вращательное движение вала электродвигателя преобразуется в поступательное движение штанг с помощью передачи винт-гайка.

САУ триподом в качестве входных сигналов имеет требуемые углы поворота платформы, а в качестве выходных сигналов – управляющие напряжения для исполнительных электроприводов. Вследствие того, что параметры самого трипода (момент инерции платформы, например), а также параметры внешней нагрузки на секцию могут меняться в широких пределах, САУ должна быть адаптивной.

Моделирование исполнительных двигателей САУ. Структура используемой системы управления исполнительным двигателем постоянного тока изображена на рис. 2.

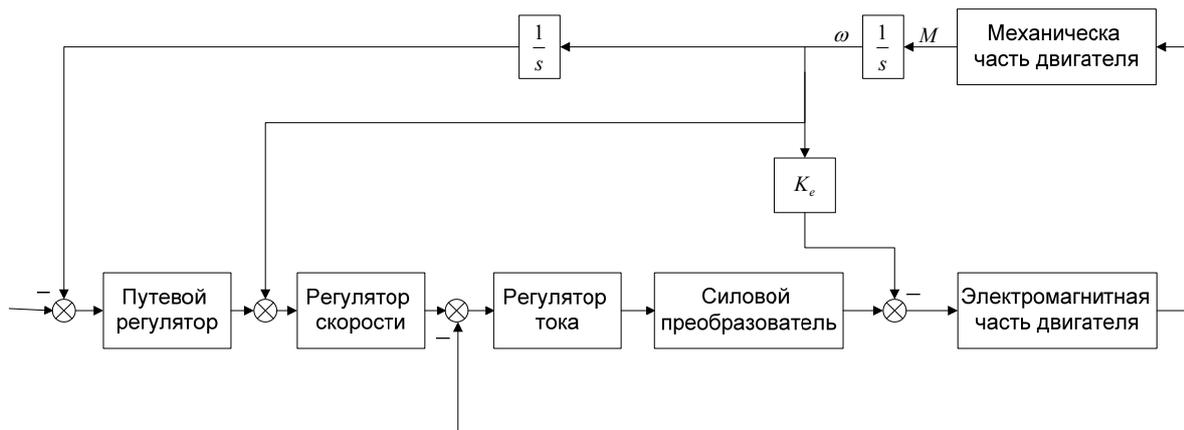


Рис. 2. Структура системы управления электроприводом

Исследование системы управления, представленной на рис. 2, выполнено средствами системы Simulink программного комплекса MatLab. Некоторые результаты исследования представлены на рис.3.

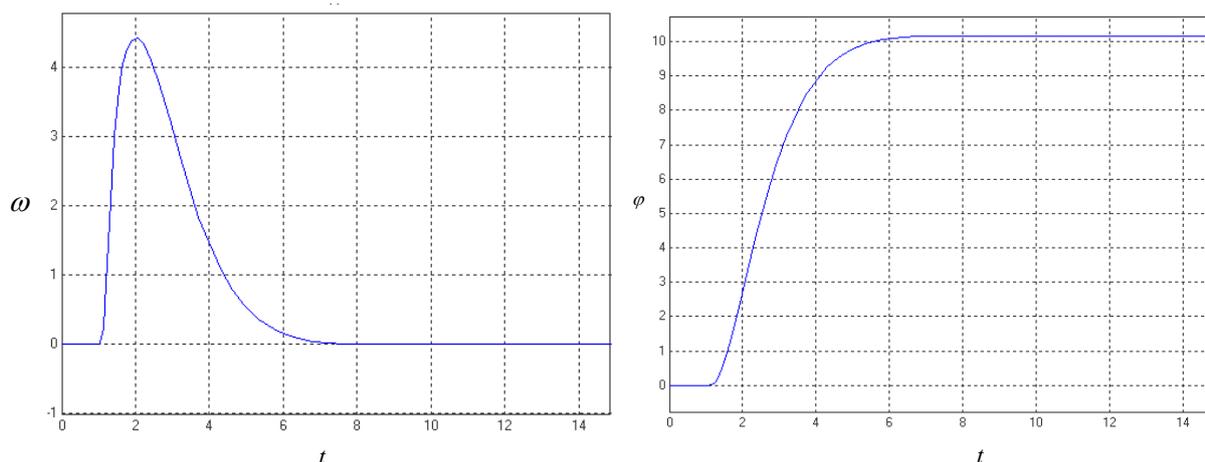


Рис. 3. Переходные процессы в системы управления электроприводом

Все регуляторы, указанные на рисунке 2, представляют собой пропорционально-интегральные звенья; силовой преобразователь и электромагнитная часть двигателя моделируются аperiodическими звеньями; модель механической части двигателя связывает ток, скорость вращения и момент на валу электродвигателя.

Адаптивная САУ на основе эталонной модели. Структура адаптивной САУ представлена на рис. 4.

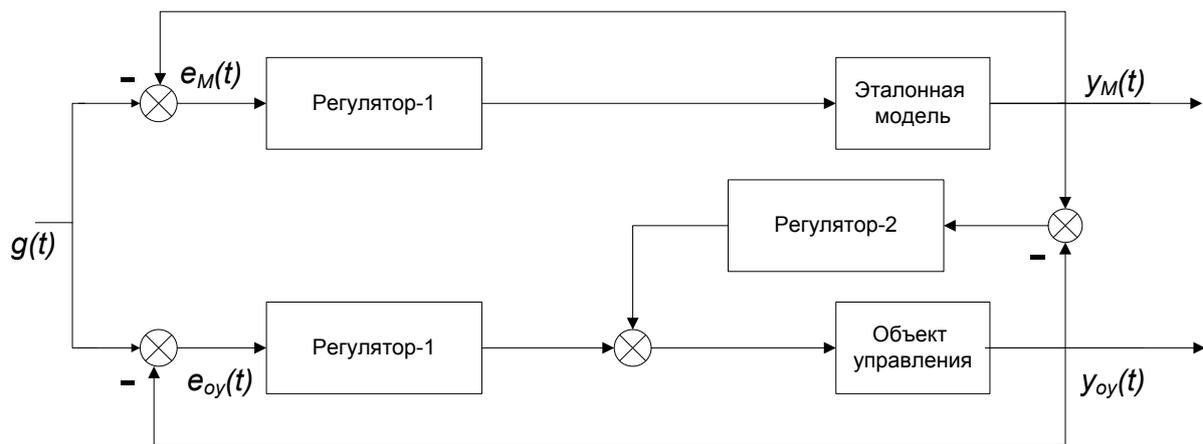


Рис. 4. Структурная схема САУ с эталонной моделью

САУ состоит из двух контуров, в один из которых входит эталонная модель объекта, а в другой контур – сам объект управления. Оба контура имеют обратную связь. Управляющее воздействие подается как на объект управления, так и на его модель. Регуляторы модели и объекта (регуляторы-1) полностью идентичны. Выходной сигнал модели и объекта вычитаются, и рассогласование компенсируется регулятором-2, выходное значение которого складывается с выходом регулятора-1 объекта управления.

Исследование САУ с эталонной моделью также выполнено средствами системы Simulink программного комплекса MatLab.

Нейросетевая система управления. В работе рассматривается также построение нейросетевой адаптивной САУ. Известны два основных метода создания нейросетевых САУ: прямой метод (в качестве регулятора используется непосредственно нейронная сеть); косвенный метод (нейронная сеть выступает в качестве модели объекта управления) [2].

Косвенный метод является основным при построении нейросетевых САУ. В качестве нейросетевой модели объекта управления в этом случае используется инверсная модель, передаточная характеристика которой является обратной по отношению к передаточной функции модели объекта управления. При включении в контур управления инверсной модели она компенсирует выход объекта управления таким образом, что ошибка регулирования становится минимальной. Если, например, объект управления описывается апериодическим звеном с передаточной функцией $\frac{k}{Ts+1}$, то инверсная модель такого

объекта определяется передаточной функцией $\frac{Ts+1}{k}$. Таким образом, при последовательном включении этих звеньев в контур управления, общая передаточная функция становится равной единице.

Общая схема нейросетевой САУ представлена на рис. 5.

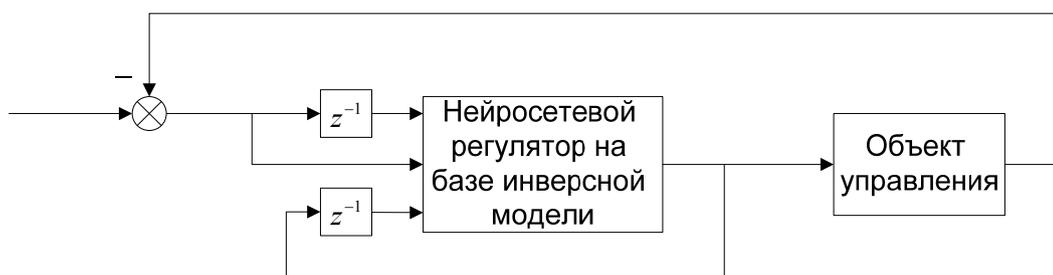


Рис. 5. Структура нейросетевой САУ

Нейросетевой регулятор на рисунке 5 представляет собой нейронную сеть, которая обучается на инверсной модели объекта управления.

Заключение. В работе показано, что имеется несколько принципиально различных подходов к построению адаптивных САУ секцией манипулятора типа «хобот». Рассмотрены два метода: метод синтеза САУ на основе эталонной модели и метод на основе нейронных сетей. Выполнено моделирование обеих систем средствами системы Simulink программного комплекса MatLab. Многочисленные эксперименты с этими моделями показали удовлетворительные результаты на различных тестовых входных сигналах.

Остается открытым вопрос устойчивости рассмотренных САУ. Сложность исследования устойчивости заключается в отсутствии единого метода исследования для таких систем, вследствие их высокой нелинейности. В САУ с эталонной моделью нелинейность заключается непосредственно в эталонной модели, описываемой системой нелинейных обыкновенных дифференциальных уравнений, а в нейросетевой САУ нелинейность вносит сама нейронная сеть.

Литература

1. Каганов Ю.Т., Карпенко А.П. Математическое моделирование кинематики и динамики робота-манипулятора типа «хобот». 1. Математические модели секции манипулятора, как механизма параллельной кинематики типа «трипод» // Наука и образование: электронное научно-техническое издание, 2009,10, 0420900025\0048 (<http://technomag.edu.ru/doc/133262.html>).

2. Методы робастного, нейро-нечеткого и адаптивного управления. Учебник / Под редакцией Н.Д. Егупова.- М.: Изд-во МГТУ им. Н.Э. Баумана, 2002.- 744 с.

ПОСТРОЕНИЕ ОРГАНИЗАЦИОННОЙ СТРУКТУРЫ ПРОЕКТА СО СПЕЦИАЛИЗИРОВАННЫМ ХАРАКТЕРОМ РАБОТ

Мицик О.М.,

Научный руководитель: Берчун Ю.В.

МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Российская Федерация

COMPOSITION OF A PROJECT ORGANIZATION STRUCTURE WITH SPECIFIC WORK TYPES

Mitsik O.M.

supervisor Berchun U.V.

BMSTU, Moscow, Russian Federation

Аннотация

Настоящая работа посвящена задаче поиска оптимальной организационной структуры проекта со специализированным характером работ. Особое внимание уделяется подзадаче распределения работ проекта по единицам организационной структуры с учетом их компетенции.

Abstract

This work is devoted to the problem of finding the optimal organization structure for projects with specific work types. The main attention is drawn to a subtask of the distribution project works among organization structure units considering their competence levels.

Задачи оптимизации организационных структур в управлении социально-экономическими системами являются весьма актуальными [1]. В течение последних лет вопросы построения организационных структур в системах проектного управления рассматривались в ряде отечественных и зарубежных работ [2,3]. Для решения задач без учета специализации единиц организационной структуры существует достаточно эффективный аппарат с элементами оптимизации. Однако без должного внимания остаются задачи построения оптимальной организационной структуры для проектов со специализированным характером работ.

В работе используются следующие понятия:

- структура работ проекта (WBS), представляется декомпозиционной иерархической структурой работ, каждая работа характеризуется набором параметров (объемом работы, ее продолжительностью и информационной сложностью);
- организационная структура проекта (OBS), представляющая собой управленческую команду проекта (набор менеджеров), распределенных по иерархической структуре (уровням управления);
- коэффициент информационной сложности (КИС) работы, повышающий или понижающий объем управленческой информации, требуемой для выполнения работы;

– интервал информационной мощности по каждому из уровней управления (ресурс уровня управления для выполнения работы);

– вектор информационной мощности каждого менеджера.

Исходные данные проекта:

- объем трудозатрат в часах T_i , $i=1,2,\dots, n$ для выполнения i -й работы;
- длительности работ;
- векторы информационной мощности менеджеров;
- структура работ проекта (WBS);
- интервалы информационной мощности M_k , $k=1,2,\dots,N$.

Условные обозначения:

N – количество уровней проекта WBS;

$a_{i,k}$ – i -я работа k -го уровня WBS, $k=1,2,\dots,N$;

$t(a_{i,k})$ – длительность ведения работы $a_{i,k}$;
 $\eta_{i,k}$ – количество работ $(k+1)$ – го уровня WBS, подчиненных работе $a_{i,k}$;
 $TP(a_{i,k})$ – трудоемкость выполнения работы $a_{i,k}$;
 $OTP(a_{i,k})$ – средняя производительность менеджера при выполнении работы $a_{i,k}$;
 $КИ(a_{i,k})$ – количество исполнителей задачи $a_{i,k}$;
 $Q^{(m)}$ – совокупность подразделений OBS на m – ом уровне управления,
 $1 \leq m \leq M$, M – количество иерархических уровней OBS проекта;
 $U^{(m)}$ – суммарная мощность всех элементов OBS, входящих в $Q^{(m)}$;
 $x_s^{(m)}$ – значение для менеджера, (1, если он участвует в проекте и 0 – если нет);
 Q – организационная иерархическая структура OBS: $Q = \bigcup_m Q^{(m)}$;
 $KU_{\min}^{(m)}$ – минимальное число менеджеров на m - ом уровне OBS;
 KU_{\min} – минимальное число менеджеров, участвующих в OBS;
 KU – общее количество менеджеров, которые могут участвовать в OBS;
 U – суммарная мощность менеджеров, которые могут участвовать в OBS;
 $КИС^{(m)}$ – суммарное значение информационной потребности для m - го уровня OBS
(определяется по данным WBS^{\min});
 $КИС(a_{i,n})$ – коэффициент информационной сложности по решению задачи $a_{i,n}$
 $g_{\min}^{(m)}, g_{\max}^{(m)}$ – нижняя и верхняя грани фильтра на m - ом уровне OBS;
 Π_i – производительность труда i -го исполнителя.

Требуется определить минимальное число руководителей $KU_{\min}^{(m)}$ для каждого уровня OBS, $1 \leq m \leq M$, обеспечивающее реализацию проекта и минимальную суммарную стоимость работ менеджеров, участвующих в проекте

Таким образом, требуется найти минимумы целевых функций

$$KU_{\Sigma} \rightarrow \min, \quad (1)$$

$$\sum_{m=1}^M U^{(m)} \rightarrow \min, \quad (2)$$

при ограничениях

$$\sum_{s=1}^{KU_{\min}^{(m)}} U_s^{(m)} \geq КИС^{(m)}. \quad (3)$$

Задача (1-3) является двухкритериальной задачей дискретного линейного программирования. Оптимальное решение по двум критериям находится как компромиссный оптимум. Целевой функцией для компромиссного критерия является

$$\Phi = \alpha KU_{\Sigma} / KU + (1 - \alpha) \sum_{m=1}^M U^{(m)} / U \rightarrow \min \quad (4)$$

Значение параметра α выбирается в зависимости от того, какой из критериев имеет большую относительную значимость в конкретной задаче.

Для построения оптимальной OBS предлагается подход, включающий следующие этапы:

1) расчет управленческой информации по каждой работе WBS. Определение коэффициентов информационной сложности работ по нормативам и базам статистических данных, по проектам-аналогам или с помощью экспертов.

2) построение минимальной структуры работ проекта WBS^{min} на основе WBS проекта с учетом заданных интервалов информационной мощности на каждом из уровней управления.

3) построение оптимальной организационной структуры проекта OBS^{min} на основе построенной минимальной структуры работ WBS^{min} и векторов информационной мощности каждой из штатных единиц элемента управления. Построение оптимальной организационной структуры осуществляется путем нахождения значений $KU_{min}^{(m)}$ методами линейного программирования и с использованием генетического алгоритма, с учетом ограничений $KIC^{(m)}$, полученных на основе WBS^{min} .

В процессе построения базы информационной сложности работ для каждой работы $a_{i,n}$ определяется количество исполнителей, необходимое для выполнения работы, зависящее от предполагаемой длительности работы, ее суммарной трудоемкости и средней производительности одного трудового ресурса:

$$KI(a_{i,n}) \cdot ЧОТР(a_{i,n}) = \sum_{k=1}^{KI(a_{i,n})} \Pi_k = \frac{TP(a_{i,n})}{t(a_{i,n})}, \quad (5)$$

Примем за основу модель КИС руководителя для рассматриваемой задачи в виде зависимости от производительности труда исполнителей и количества исполнителей, привлеченных к решению задачи. Коэффициент информационной сложности будем определять из зависимости типа Кобба-Дугласа:

$$KIC(a_{i,n}) = KI(a_{i,n})^\gamma \cdot ЧОТР(a_{i,n})^\beta. \quad (6)$$

Параметры γ, β следует определять из эмпирических данных для конкретных приложений.

Каждая, входящая в WBS i -я работа на k -ом уровне иерархии $a_{i,k}$, $1 \leq k \leq N$, имеет структуру:

$$a_{i,k} = \bigcup_q a_{q,k-1}, \quad \forall q: 1 \leq q \leq \eta_{i,k}. \quad (7)$$

т.е. объем работы k -го уровня равен сумме объемов работ $(k+1)$ -го уровня, подчиненных работе $a_{i,k}$.

Строится новая структура работ WBS^{min} , которая называется минимальной и которая удовлетворяет следующим условиям:

1) число уровней WBS^{min} , также, как и в организационной структуре OBS равно M ;

2) для i -й работы на m -ом уровне WBS^{min} выполняется

$$g_{min}^{(m)} \leq KIC(a_{i,k}) \leq g_{max}^{(m)}, \quad 1 \leq k \leq N, 1 \leq i \leq 3^{(k)}, \quad (8)$$

где: $\eta^{(k)}$ – число работ дерева на k -ом уровне;

значения $g_{min}^{(m)}, g_{max}^{(m)}$ задают границы интервалов информационной мощности на m -ом уровне управления;

3) для каждого k -го уровня иерархии, $1 \leq k \leq M$, имеет место соотношение

$$\sum_{i=1}^{\eta_k} t(a_{i,k+1}) = t(a_{i,k}), \quad (9)$$

то есть сумма времени на выполнение всех подчиненных работ $k+1$ -го уровня равно суммарному времени, затрачиваемому на решение задачи k -го уровня.

Поиск оптимума целевой функции (4) при ограничениях (3) находится методами линейного программирования [6]. В качестве начального решения ЗЛП использовалось решение, в котором были назначены руководители с наибольшими значениями своей

информационной мощности и при условии выполнения всех неравенств системы ограничений (3).

Поскольку по условию существует набор менеджеров, обеспечивающих работу проекта, то такая задача имеет решение. Обозначим это решение в виде:

$$X_{3ЛП}^{opt} = (X_1^{(1)}, X_2^{(1)}, \dots, X_{k_1}^{(1)}, X_1^{(2)}, X_2^{(2)}, \dots, X_{k_2}^{(2)}, X_1^{(M)}, X_2^{(M)}, \dots, X_{k_M}^{(M)}). \quad (10)$$

По решению (18) строится целочисленное решение, являющееся исходным для генетического алгоритма [4].

$$X_{ГЕН}^0 = (\bar{X}_1^{(1)}, \bar{X}_2^{(1)}, \dots, \bar{X}_{k_1}^{(1)}, \bar{X}_1^{(2)}, \bar{X}_2^{(2)}, \dots, \bar{X}_{k_2}^{(2)}, \bar{X}_1^{(M)}, \bar{X}_2^{(M)}, \dots, \bar{X}_{k_M}^{(M)}), \quad (11)$$

$$\text{где } \bar{X}_s^{(m)} = \begin{cases} 0, & \text{если } X_s^{(m)} = 0; \\ 1, & \text{если } 0 < X_s^{(m)} \leq 1; \end{cases} \quad m = 1, \dots, M; s = 1, \dots, k_m.$$

Размер популяции в задаче ГА определяется по формуле

$$N_0 \geq \frac{\ln(P(R_L))}{\ln\left(1 - \left(\frac{1}{2}\right)^L\right)}, \quad (12)$$

где L- размерность вектора (11), в каждой из N_0 хромосом меняется один из генов с вероятностями согласно правилу рулетки.

$$p_i = \frac{\Phi_i - \Phi_{\min}}{\sum_{j=1}^{N_0} (\Phi_j - \Phi_{\min})}, \quad (13)$$

где: $\Phi_i, i = 1, \dots, N_0$ – значения, вычисляемые для целевой функции на хромосомах;

Φ_{\min} – значение целевой функции в точке оптимума $X_{3ЛП}^{opt}$.

Алгоритм заканчивает работу при заданном количестве повторений среднего значения для популяции для нескольких последовательных шагов.

С помощью генетического алгоритма [5] находится решение, которое является целочисленным из набора с координатами $\{0;1\}$ и показывает, кого из руководителей необходимо включить в работу над проектом (значение $X_s^{(m)}=1$). При этом на множестве допустимых целочисленных решений получено решение, доставляющее минимум целевой функции (4) то есть минимум компромиссного критерия.

Таким образом определяется решение задачи по поиску минимальной организационной структуры проекта.

Литература.

1. Математические основы управления проектами. Учебн. пособие для ВУЗов, под ред. Буркова В.Н. М.: Высшая школа, 2005. – 243 с.
2. Бушуев С., Сочнев С., Цветков А. Методы управления сложными объектами на основе энтропийного подхода. – В кн.: Сб. докл. симп. «Управление проектами в переходной экономике: инвестиции, инновации, менеджмент». – М.: СОВНЕТ, 1997. С. 78 – 86.
3. [Механизмы управления организационными проектами.](#) / В.Г. Балашов, А.Ю. Заложнев, Д.А. Новиков. М.: ИПУ РАН, 2003. - 84 с.
4. Mitchell, M. An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA (1996).
5. Батищев Д.И. Генетические алгоритмы решения экстремальных задач. Под ред. Львовича Я. Е.: Учеб. пособие, Воронеж, 1995.
6. Таха Х. Введение в исследование операций: Пер. с англ. - М.: Вильямс.2001.

ПРИМЕНЕНИЕ АППАРАТА НЕЧЕТКОЙ ЛОГИКИ ПРИ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ

Марченков А.М.

Научный руководитель к.т.н. Волосатова Т.М.

МГТУ им. Н.Э. Баумана, кафедра РК6, Москва, Россия

APPLICATION OF THE INSTRUMENT OF FUZZY LOGIC IN THE SEGMENTATION

Marchenkov A.M.

The scientific chief: Ph.D. senior lecturer Volosatova T.M.

Department RK6 BMSTU, Moscow, Russia

Аннотация

Приведен обзор цифровой обработки изображения и задачи сегментации, понятий и методов нечеткой логики. Предложен метод решения задачи сегментации с использованием аппарата нечеткой логики. Приведен интерфейс программной реализации. Проиллюстрировано решение поставленной задачи.

Abstract

An overview of digital image processing and segmentation tasks, concepts and methods of fuzzy logic is given. A method for solving the problem of segmentation using the apparatus of fuzzy logic is suggested. An interface software implementation is given. The solution of the problem is illustrated.

Одна из основных трудностей в создании хороших программ распознавания объектов состоит в определении того, какие пиксели нужно распознавать, а какие – игнорировать. Термин «сегментация» употребляется для обозначения многих процессов, которые связаны общей идеей: получением компактного представления полезного содержимого объекта. К сожалению, доступной теории сегментации не существует, и не последнюю роль в этом сыграло то, что лишь из конкретного приложения определяется, что на объекте представляет практический интерес.

По сути своей, сегментация – это попытка определить, какие компоненты набора входных данных естественно связать вместе. Такая задача является задачей кластеризации. Кластеризацию можно проводить двумя способами:

1) Разбиение. Большой набор данных нужно разделить по определенным представлениям об ассоциации элементов набора. Обычно разбиение проводят по частям, которые удобны для используемой модели. Можно, например, разбить исходное изображение на когерентные области цвета и текстуры. Также можно разбить изображение на большие пятна, состоящие из когерентных областей цвета, текстуры и движения.

2) Группировка. В этом случае из определенного набора различных информационных изображений необходимо собрать наборы элементов, значимых с точки зрения используемой модели. Наличие различных эффектов, например затенения, означает, что некоторые компоненты изображения, которые принадлежат одному объекту, на изображении могут быть разнесены. Примером группировки является сбор токенов, в результате которого формируется линия [1].

Ключевым вопросом является определение представления, которое подходит для поставленной задачи. Нужно выбрать критерий, согласно которому схема сегментации будет определять, какие пиксели нужно группировать вместе. Самый плодотворный источник необходимых для этого знаний – это человеческая зрительная система, решающая эту задачу в общей форме и способная легко подбирать такой критерий.

В случае, когда критерием сегментации изображения является яркость или цвет, для нахождения пикселей, относящихся к объекту, можно использовать аппарат нечеткой логики [3], так как яркость пикселей объекта не является строго заданной величиной, а

распределена в некотором диапазоне. Это может быть вызвано целым рядом различных причин, например: текстурой самого объекта, различным уровнем и качеством освещенности объекта, шумами и т.д. В качестве примера можно привести изображение зашумленного силуэта хромосомы (рис. 1) и гистограммы (рис. 2), соответствующей этому изображению [2].

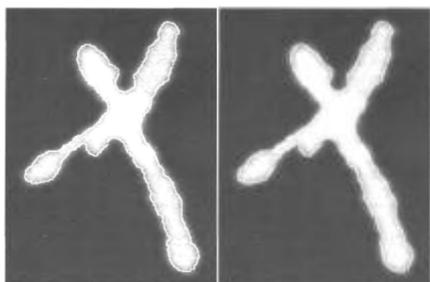


Рис.1. Изображение зашумленного силуэта хромосомы

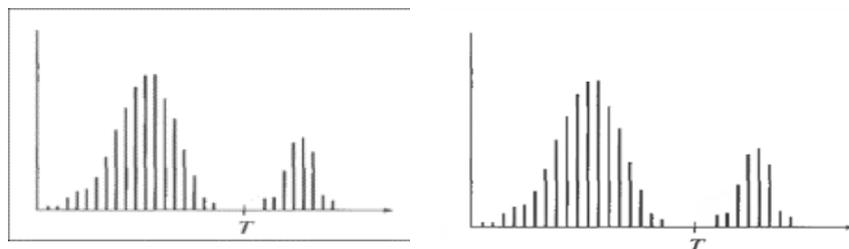


Рис.2. Гистограмма

Для цветных изображений пиксель описывается тремя базовыми цветами системы RGB, и соответственно нужно рассматривать диапазон по каждому из них.

Для описания системы с помощью нечеткой логики используются два основных понятия: лингвистическая переменная и терм лингвистической переменной. Переменная в нечеткой логике соответствует одному из параметров, характеризующих систему, причем все множество «четких» значений этой переменной будет образовывать ее область определения.

Терм переменной является «нечетким» значением, он обозначается фразой, которая характеризует одно из характерных состояний параметра системы.

Для задачи сегментации за лингвистическую переменную примем цвет объекта. А терминами к ней будут являться: красный, зеленый, синий.

Терм представляет собой нечеткое множество, определяющееся парой, состоящей из элемента из области определения переменной и числа из отрезка $[0,1]$. Таким образом, он определяет степень истинности принадлежности яркости пикселя к характерному значению переменной, которой назван этот терм.

Характеристическую функцию или функцию принадлежности нечеткого множества можно задать несколькими способами. Первый способ – таблицей, где каждому элементу глобального множества будет поставлено значение характеристической функции; второй способ – задание функции принадлежности графически. В общем случае характеристическая функция может иметь любой вид, однако к настоящему моменту сформировалось понятие о стандартных функциях: Z – функция, S – функция, П – функция, Л – функция (рис. 3) [4].

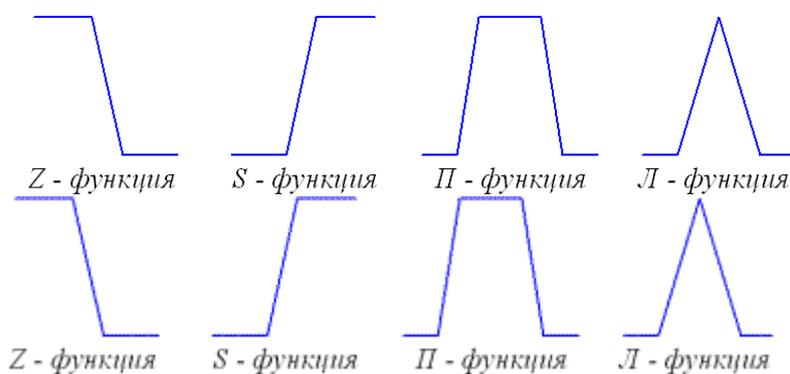


Рис.3. Стандартные функции принадлежности нечеткого множества.

Для сегментации изображений наиболее логично выбрать П – функцию, так как некоторый диапазон около заданного значения яркости практически не отличим человеческим глазом, и его можно принять за единицу.

Для улучшения сегментации изображения рассмотрим несколько вариантов задания продукционных правил, то есть правил поведения системы.

Если в искомой яркости преобладает какая-то из составляющих R, G или B, то значение, при котором пиксель будет определяться как принадлежащий искомому объекту, должно быть попадать в более узкий диапазон, чем две другие составляющие. Например, при преобладании красной составляющей правила могут выглядеть так, как представлено на рис. 4.

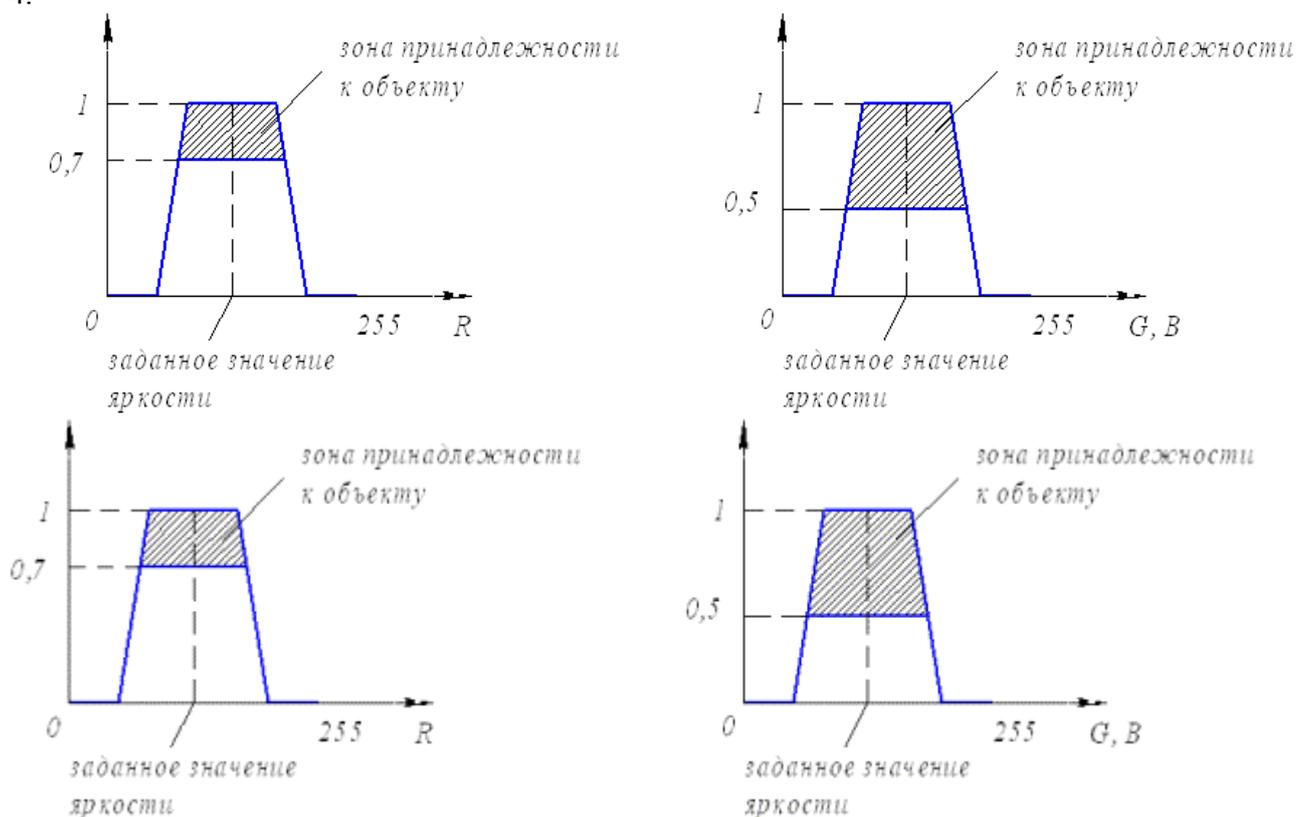


Рис.4. Продукционные правила.

Такое же правило можно использовать при преобладании двух составляющих, ограничивая их более узким диапазоном, а также если нельзя выделить преобладающую составляющую, то есть при градации черного цвета, в этом случае такое ограничение налагается на все составляющие.

Разработано программное обеспечение для сегментации изображения, в котором реализован и описанный выше метод, графический интерфейс данной программы представлен на рис. 5.

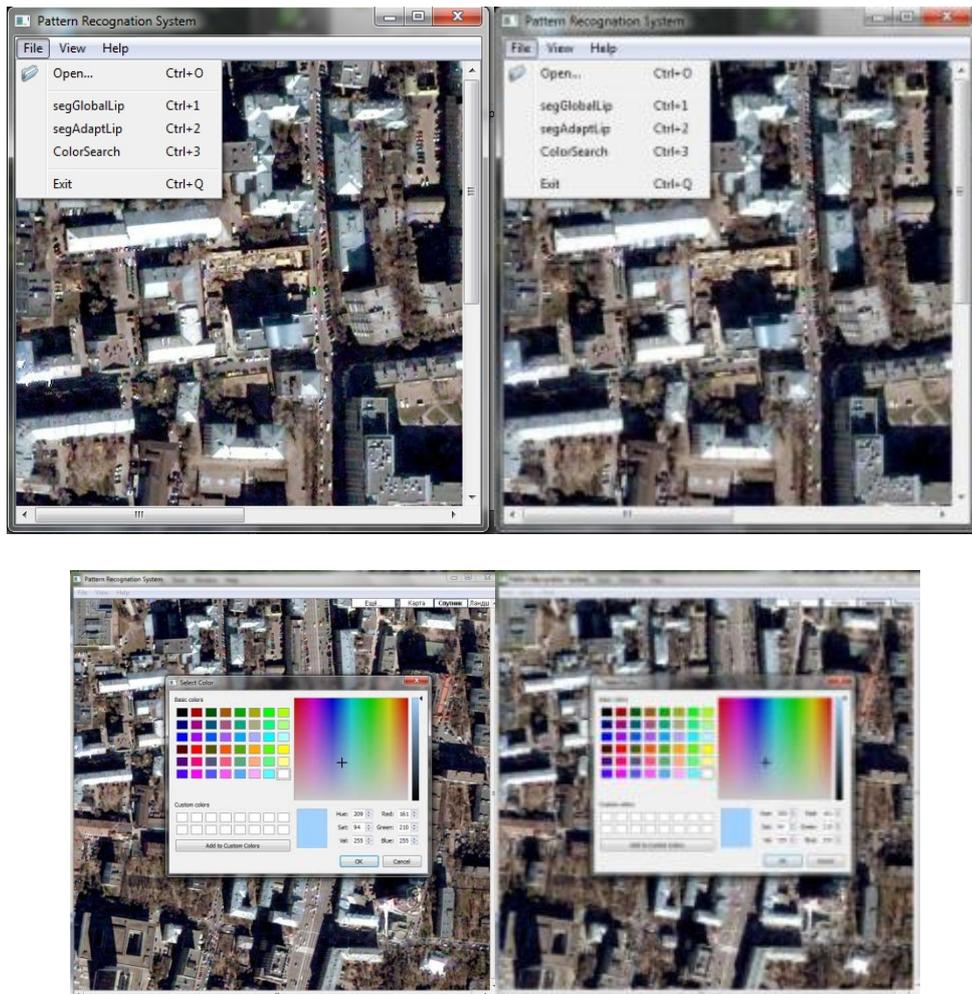


Рис.5. Интерфейс программы.

После открытия обрабатываемого изображения в окне программы можно выбрать метод поиска цвета из меню File или нажатием комбинации клавиш Ctrl + 3. Далее пользователю предоставляется выбор задания искомого цвета, либо кликнув по изображению, либо задав вручную с помощью появляющегося диалога выбора цвета. Результаты работы представлены на рис. 6, пиксели фона закрашены, а те, что отнесены к объекту, остались без изменения.

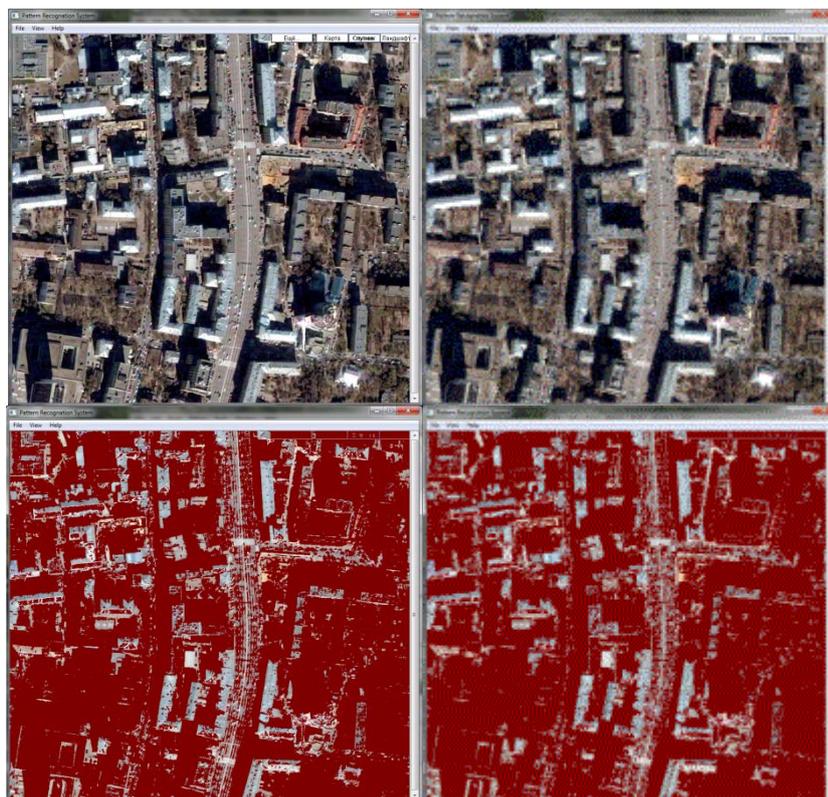


Рис. 6 Исходное и полученное изображения

В данный момент ведется работа по внедрению методов улучшения качества исходных изображений для получения более качественного результата, и по дальнейшей обработке, полученной в результате сегментации информации.

Литература

1. Д. Форсайт, Ж. Понс – Компьютерное зрение. Современный подход. : перевод с английского – М. – Издательский дом «Вильямс», 2004. – 928с., Москва – Санкт-Петербург – Киев, 2004.
2. Р. Гонсалес, Р. Вуде – Цифровая обработка изображений. : перевод с английского под редакцией П. Н. Чочиа – «Техносфера», Москва, 2005.
3. Л. Заде – Понятие лингвистической переменной и его применение к принятию приближенных значений – «Знание», Москва, 1976.
4. Волосатова Т.М., Жиряков С.М., Рогозин О.В., Филиппов М.В. - Применение аппарата нечеткой логики в построении контроллеров - "Мехатроника, автоматизация, управление" №1, Москва, 2009.

РАСПОЗНАВАНИЕ ГЕОМЕТРИЧЕСКИХ ПРИМИТИВОВ В СИСТЕМЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Афиногенов Е.И., Александров Д.А.

Научный руководитель: к.т.н, доцент Волосатова Т.М.
Кафедра «САПР» (РК6) МГТУ им. Н.Э.Баумана, Москва, Россия

RECOGNITION OF GEOMETRIC PRIMITIVES IN COMPUTER VISION SYSTEM

Afinogenov E.I., Alexandrov D.A.

Scientific chief: PhD, associate professor Volosatova T.M.
CAD/CAE Department (RK6) of BMSTU, Moscow, Russia

Аннотация

Рассматривается задача распознавания элементарных геометрических примитивов (типов поверхностей) в трехмерном облаке точек. Предлагаемая методика ее решения основана на вычислении локальных особенностей относительного расположения точек (PFH – Point Feature Histogram), и сопоставлении конкретных значений с предварительно сформированной базой данных известных геометрических примитивов. Для классификации применяется метод опорных векторов (SVM — Support Vector Machines).

Abstract

This article refers to the problem of structural analysis of 3D point clouds and recognition of geometrical primitives. A new approach called Point Feature Histogram (PFH) is used to obtain local surface descriptors and is combined with SVM (Support Vector Machines) algorithm to convert numerical descriptor into surface label.

Введение

В настоящее время сегментация и интерпретация трёхмерных сцен в робототехнике становится важной темой исследований. Распознавание и реконструкция пространственных объектов играют важную роль в решении задач манипуляции и навигации.

Схема действия предлагаемой системы компьютерного зрения

1. По изображениям, полученным с помощью двух камер, рассчитываются координаты точек наблюдаемого объекта в пространстве.
2. Далее необходимо структурировать данное облако точек для упрощения манипуляций и эффективного поиска ближайших соседей точек облака. В данном случае предлагается использовать триангуляцию Делоне.

Триангуляцией называется планарный граф, все внутренние области которого являются треугольниками (рис. 1).

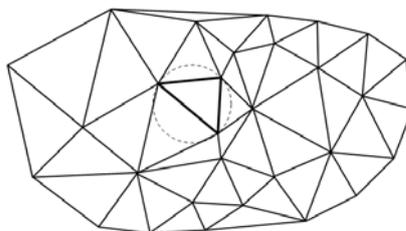


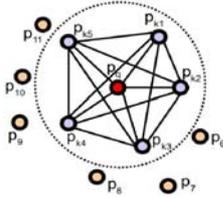
Рис. 1

Говорят, что триангуляция удовлетворяет условию Делоне, если внутри окружности, описанной вокруг любого построенного треугольника, не попадает ни одна из заданных точек триангуляции.

3. Расчет вектора нормали в каждой точке.

4. Для определения принадлежности точки к какому-либо типу поверхности (плоскость, сфера, цилиндр, конус и т.д.) используется механизм PFH (Point Feature Histogram), представляющий собой гистограмму распределения векторов нормалей в каждой точке облака, лежащей в окрестности целевой точки.

Для получения гистограммы PFH из множества точек, лежащих в данной окрестности, формируются все возможные пары.



$$\frac{k(k-1)}{2} \text{ пар, где } k \text{ – число точек в окрестности.}$$

Рис. 2

Для каждой пары точек рассчитываются 3 угловые величины (α, ϕ, θ) , характеризующие взаимное расположение нормалей к поверхности в точках:

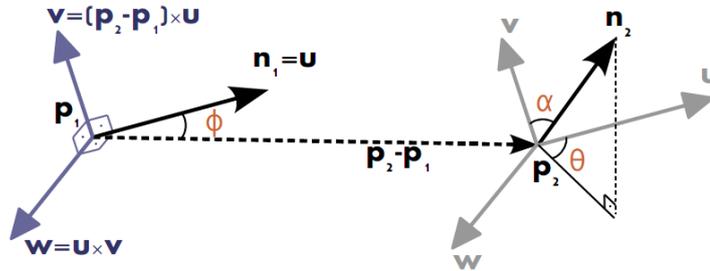


Рис. 3

$$\begin{aligned} \alpha &= v \cdot n_2 \\ \phi &= (u \cdot (p_2 - p_1)) / \|p_2 - p_1\|_2 \\ \theta &= \arctan(w \cdot n_2, u \cdot n_2) \end{aligned}$$

Где $\{u, v, w\}$ – ортонормированная система (Darboux frame), привязанная к одной из данной пары точек (к точке, нормаль которой образует меньший угол с вектором $p_2 - p_1$).

$$u = n_1 \quad v = (p_2 - p_1) \times u \quad w = u \times v$$

Гистограмма PFH рассчитывается с учетом значений, принимаемых тройкой (α, ϕ, θ) для каждой пары точек из рассматриваемой окрестности.

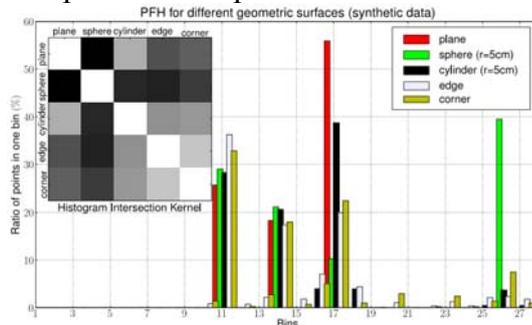


Рис. 4

Показано, что PFH является универсальным дескриптором точки поверхности в пространстве, инвариантным к повороту, перемещению и в большой степени к масштабированию и плотности облака точек.

Экспериментальная часть

В данной работе применяется размерность дескриптора равная 8. Для исследования была сформирована поверхность, включающая плоский, цилиндрический и сферический фрагменты. Поверхность генерируется по маске, с добавлением шума по координате глубины. На первом этапе рассчитываются дескрипторы точек и маркируются в соответствии с маской. На втором этапе происходит обучение SVM-классификатора по полученным соответствиям «дескриптор – тип поверхности». На третьем этапе повторяется генерация нового набора точек по маске, и тип поверхности определяется при помощи классификатора. В результате рассчитывается число правильных и неправильных ответов классификатора, и формируется графическое представление определенного типа поверхности.

Входные данные классификатора

Для генерации точек используется маска (рис. 5):

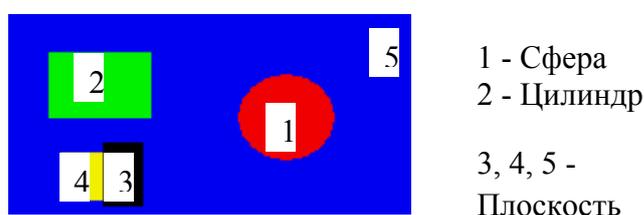


Рис. 5. Маска типов поверхности

По маске формируется облако точек с шумом в +/- 5% в направлении оси Z. Размеры облака – 200 точек * 100 точек = 20 000 точек.

Обучение классификатора

Для обучения классификатора используется полный набор дескрипторов, рассчитанный для каждой из 20 000 точек с радиусом 4 точки.

Поскольку пространство входных параметров имеет довольно большую размерность, и различные типы поверхности дают свои характерные соотношения значений, можно предположить, что выборка окажется разделимой. В качестве функции ядра используется радиально-базисная функция. Классификатор обучается различать эти 3 типа поверхностей. Обучение на 20 000 входных векторах занимает порядка 4 секунд. Построенная модель включает в себя 1468 опорных векторов.

Тестирование классификатора

Тестирование проводилось на новом облаке точек, сгенерированном по старой маске. Результат представлен в виде процента верных и неверных ответов классификатора. Радиус для формирования дескриптора локальных особенностей поверхности выбран равным 4 точкам, т.к. он дал оптимальный результат.

Получены следующие результаты (рис. 6):

| | Число точек | Процент |
|-------------------------|-------------|---------|
| Неудачное распознавание | 530 | 2,65 % |
| Успешное распознавание | 19 470 | 97,35 % |
| Итого | 20 000 | 100 % |

Рис. 6. Результаты теста классификатора

В графической интерпретации (рис. 7):

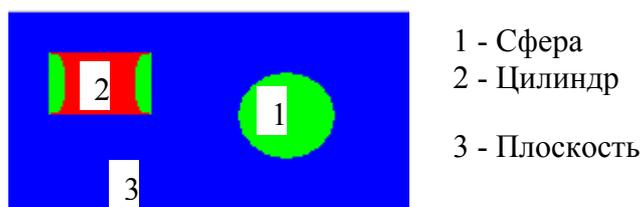


Рис. 7. Результаты теста классификатора

Заключение

Проведенные исследования позволяют предположить, что применение данной методики для обработки плотных облаков точек, полученных при сопоставлении стереоизображений, даст приемлемый результат.

Литература

1. Делоне Б.Н. О пустоте сферы // Изв. АН СССР. ОМЭН. 1934. №4. С. 793-800.
2. Скворцов А.В., Костюк Ю.Л. Эффективные алгоритмы построения триангуляции Делоне // Геоинформатика. Теория и практика. Вып. 1. Томск: Изд-во Том. Ун-та, 1998. С. 22-47.
3. Воронцов К. В. Лекции по методу опорных векторов, 2007
4. Gary Bradski, Adrian Kaehler «Learning OpenCV», 2008
5. Rusu R.B., Holzbach A., Beetz M. Detecting and Segmenting Objects for Mobile Manipulation.
6. Rusu R.B., Marton Z.C., Blodow N., Beetz M Persistent Point Feature Histograms for 3D Point Clouds
7. MITRA N. J., NGUYEN A., GUIBAS L.: Estimating surface normals in noisy point cloud data.

ПРОГНОЗИРОВАНИЕ УРОВНЯ ГЛЮКОЗЫ В КРОВИ С ПОМОЩЬЮ NARX-СЕТЕЙ

Сабуров А.А., Чернецов С.А.

д.ф.-м.н., проф. Карпенко А.П.

МГТУ им. Н.Э.Баумана, кафедра РКБ, Москва, Российская Федерация

PREDICTION BLOOD GLUCOSE LEVEL USING NARX-NETWORKS

S.A. Chernetsov, A.A. Saburov

prof. A.P. Karpenko

Bauman Moscow State Technical University, Moscow, Russian Federation

Аннотация

Рассматривается применение искусственных нейронных сетей, реализующих модель нелинейной авторегрессии с экзогенными входами (NARX-сетей), для прогнозирования уровня глюкозы в крови больных сахарным диабетом 1 типа. Исследовано влияние различных параметров NARX-сети на точность прогноза.

Abstract

The use of artificial neural networks is considered, which employ a nonlinear autoregressive model with exogenous inputs (NARX-networks), to predict the level of glucose in the blood of Type 1 diabetic patients. The effect of various parameters of NARX-networks on the accuracy of outcome prediction is investigated.

Сахарный диабет 1 типа - это метаболическое заболевание, вызванное абсолютным дефицитом секреции инсулина и характеризующееся неспособностью организма поддерживать уровень глюкозы в крови пациента (BG - Blood Glucose) в целевом диапазоне. В настоящее время разработаны препараты (генноинженерные человеческие инсулины), которые могут симулировать действие эндогенного инсулина. Для определения оптимальных типов и доз инсулина разрабатываются системы автоматического управления (САУ) уровнем BG. Центральной проблемой при разработке таких систем является проблема синтеза алгоритмов, обеспечивающих автоматическое определение требуемых доз инсулина. Указанные алгоритмы могут быть построены на основе классической теории автоматического управления, на основе теории нечетких множеств, на основе нейронных сетей, а также на основе комбинаций указанных подходов [Ошибка! Источник ссылки не найден.].

Работа посвящена исследованию эффективности прогнозирования уровня BG с помощью нейронных сетей с архитектурой NARX, общая схема которых представлена на рис. 1.

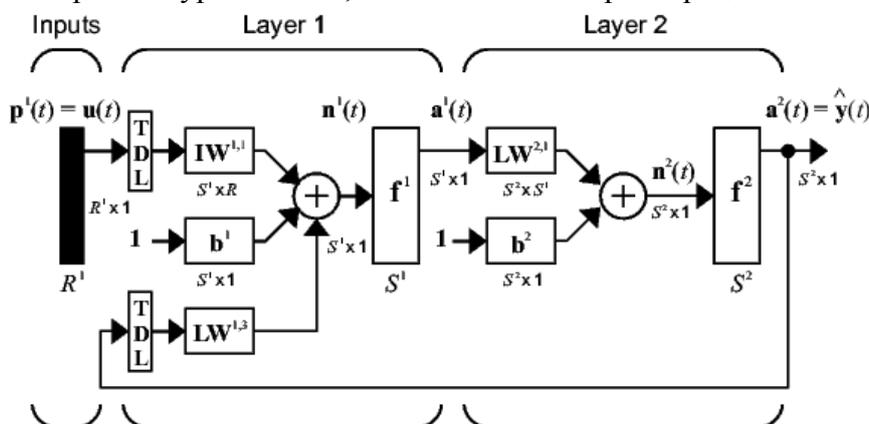


Рис.1. Общая схема NARX-сети

На рисунке 1 приняты следующие обозначения: \mathbf{IW} -матрица весов входа; \mathbf{LW} -матрица весов промежуточного слоя; \mathbf{b} -вектор смещений; \mathbf{p} -вектор входов; \mathbf{R} -количество

элементов вектора \mathbf{p} ; \mathbf{y} -выход сети; \mathbf{TDL} – линия задержки; \mathbf{f}^i -функция активации i -го слоя нейронной сети.

Функционирование NARX-сети определяется уравнением [2]

$$y(t) = f(y(t-1), y(t-2), \dots, y(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)).$$

1. Постановка задачи

Пусть

$$\Delta = (\Delta_{1-1}, \Delta_2) = (\dots, t_{-(n-1-S)}, t_{-(n-2-S)}, \dots, t_{-1-S}, t_{0-S}, t_1, t_2, \dots, t_m) -$$

- дискретная временная сетка с постоянным шагом $\delta t = 20$ мин. Здесь t_0 - текущий момент времени; $\Delta_1 = (\dots, t_{-(n-1)}, t_{-(n-2)}, \dots, t_{-1}, t_0)$ - сетка «предыстории»;

$$\Delta_{1-1} = (\dots, t_{-(n-1-S)}, t_{-(n-2-S)}, \dots, t_{-1-S}, t_{0-S}) -$$

- сетка «предыстории» с опережением, где S – величина опережения; $\Delta_2 = (t_1, t_2, \dots, t_m)$ - сетка, на которой выполняется прогноз.

Известны значения уровней ВГ и введенных пациенту доз инсулина в узлах сетки Δ_1 - векторы

$$b(\Delta_1) = (\dots, b_{-(n-1)}, b_{-(n-2)}, \dots, b_{-1}, b_0)^T, \quad i(\Delta_1) = (\dots, i_{-(n-1)}, i_{-(n-2)}, \dots, i_{-1}, i_0)^T.$$

Кроме того, известны значения компонентов вектора

$$c(\Delta_{1-1}) = (\dots, c_{-(n-1-S)}, c_{-(n-2-S)}, \dots, c_{-1-S}, c_{0-S})^T,$$

где c_{-j} - количество углеводов принятых с пищей в момент времени t_{-j} (оценивается и задается пациентом);

$$i(\Delta_{1-1}) = (\dots, i_{-(n-1-S)}, i_{-(n-2-S)}, \dots, i_{-1-S}, i_{0-S})^T,$$

где i_{-j} - уровень инсулина в момент времени t_{-j} .

Совокупность векторов $b(\Delta_1), i(\Delta_{1-1}), c(\Delta_{1-1})$ обозначим $u(\Delta_{1-1})$:

$$u = u(\Delta_{1-1}) = (b(\Delta_1), i(\Delta_{1-1}), c(\Delta_{1-1})) = (\dots, b_{-(n-1)}, \dots, b_0, \dots, i_{-(n-1-S)}, \dots, i_{0-S}, \dots, c_{-(n-1-S)}, \dots, c_{0-S})^T.$$

Использование опережения направлено на улучшение качества прогноза путем подачи на вход нейронной сети параметров, от которых в прогнозируемый момент времени зависит уровень ВГ.

Ставится задача прогноза уровня ВГ на период прогноза $T = [t_m - t_0]$, т.е. задача поиска вектора $\tilde{y} = \tilde{b}(\Delta_2) = (\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_m)^T$ [3]. Эффективность прогнозирования оценивается среднеквадратической ошибкой mse:

$$e_i = Y_i - \tilde{Y}_i; \quad mse = \frac{1}{n} \sum_{i=1}^n e_i^2.$$

2. Исследование эффективности прогноза

Для исследования использовалась программная система MatLab версии 7.9. Для ускорения подготовки и проведения экспериментов средствами MatLab разработан программный комплекс, который позволяет проводить эксперименты в пакетном режиме, определяя влияние различных параметров на эффективность прогнозирования. Также в разработанном программном комплексе предусмотрена функция построения графиков результатов экспериментов в зависимости от варьируемых параметров.

Исследование выполнено при варьировании следующих величин: значение опережения; количество нейронов в промежуточном слое сети; длины линии задержки; алгоритма обучения сети. При обучении всех рассматриваемых нейронных сетей в качестве обучающей выборки использовались данные за 5 дней (400 отсчетов); в качестве тестовой выборки - данные за 2 дня (82 отсчета).

1⁰ Варьирование величины опережения. Рассмотрены величины опережения в диапазоне от 1 до 6. Результаты исследования показали, что использование опережения в диапазоне [1:3]

оказывает положительный эффект на результаты прогноза – путем варьирования величины опережения удалось добиться снижения среднеквадратичной ошибки прогноза на ~25%. Дальнейшее увеличение опережения снижает качество прогноза.

2⁰ *Варьирование количества нейронов промежуточного слоя сети.* Показано, что лучшие результаты прогноза обеспечивает количество нейронов в промежуточном слое, равное 5. Дальнейшее увеличение количества нейронов не приводит к видимому улучшению качества прогноза, но усложняет процесс обучения нейронной сети.

3⁰ *Варьирование длины линии задержки.* Рассмотрены длины линий задержки в диапазоне [1:5]. Общая схема NARX-сети с длиной линии задержки равной n , представлена на рис. 2.

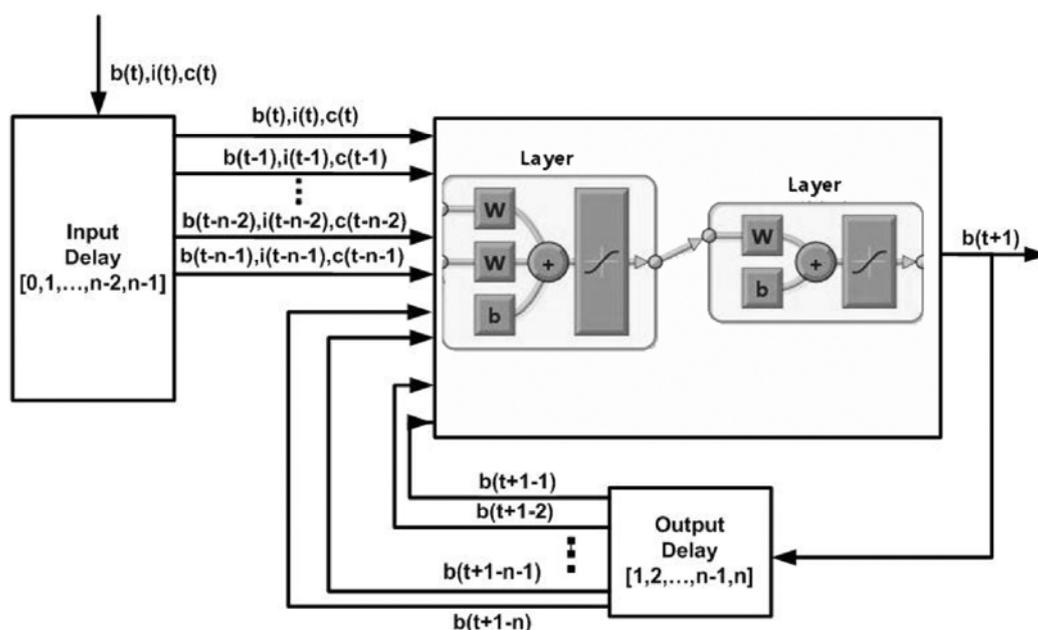


Рис. 1. Схема NARX-сети с длиной линии задержки равной n

Результаты исследования показывают, что лучшие результаты достигаются при длине линии задержки, равной 2. Дальнейшее увеличение данного параметра ведет к усложнению обучения сети и ухудшению качества прогноза.

4⁰ *Варьирование алгоритма обучения сети.* Рассмотрены следующие алгоритмы обучения: алгоритм градиентного спуска с возмущением и адаптацией параметра скорости настройки (Gradient descent with momentum and adaptive learning rate backpropagation - GDX); алгоритм Левенберга-Марквардта (Levenberg-Marquardt).

Исследование показало, что значительно лучшие результаты прогнозирования показывают NARX-сети, обученные методом градиентного спуска. Прогнозирование с помощью сети, обученной алгоритмом Левенберга-Марквардта, дает значительное количество абсолютно неприемлемых значений среднеквадратичной ошибки.

Один из лучших результатов прогнозирования с помощью NARX-сети приведен на рис. 3. Использована сеть со следующими параметрами: опережение -2; количество нейронов -5; длина линий задержки -2; алгоритм обучения – алгоритм градиентного спуска.

Заклучение

Результаты исследования показывают, что более высокая точность прогноза достигается на NARX-сетях, обучаемых алгоритмом градиентного спуска с количеством нейронов в промежуточном слое, не превышающим пяти. Использование опережения оказывает положительный эффект на результат прогноза при значениях не более трех.

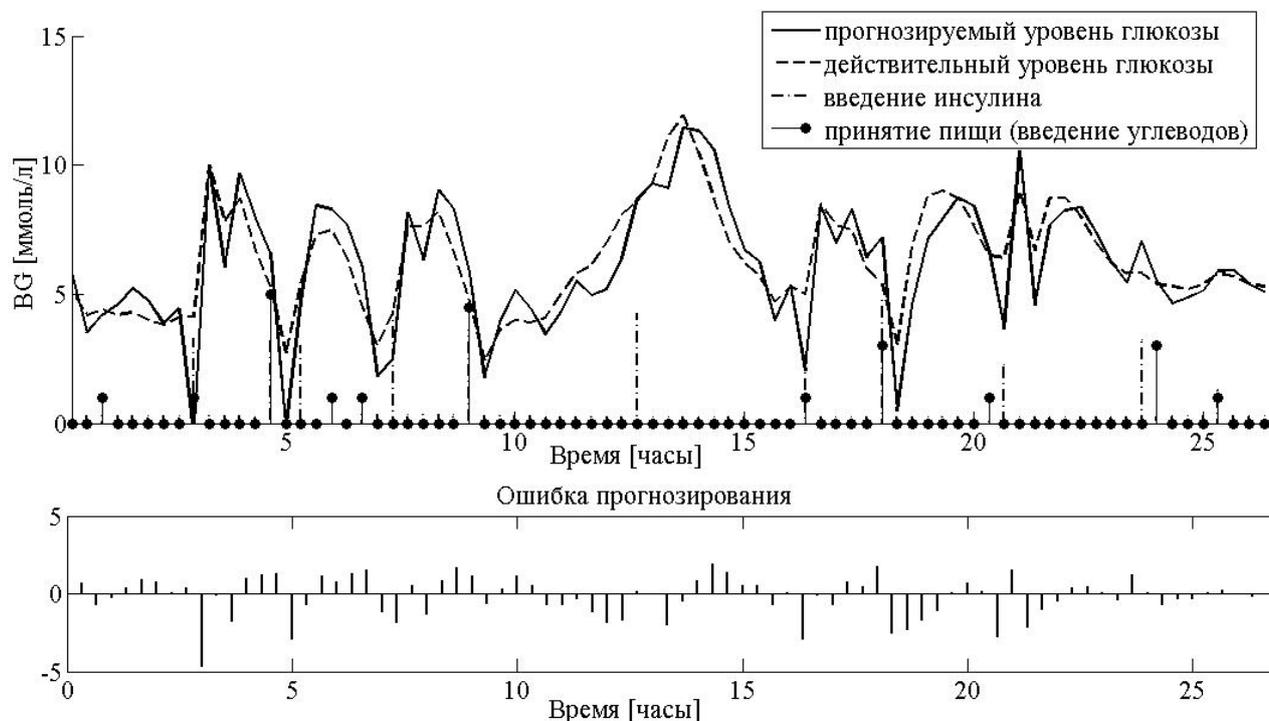


Рис. 2. Пример прогнозирования уровня ВГ с помощью NARX-сети

Литература

1. Гоменюк С.М., Емельянов А.О., Карпенко А.П., Чернецов С.А. Методы прогнозирования оптимальных доз инсулина для больных сахарным диабетом I типа. Обзор // Наука и образование: электронное научно-техническое издание, 2009, 4. (<http://technomag.edu.ru/doc/119663.html>)
2. Справочная информация для программного комплекса Matlab 7.9.0 (R2009b).
3. Емельянов А.О., Чернецов С.А. Нейросетевое прогнозирование уровня глюкозы в крови для больных инсулинозависимым диабетом // XII Всероссийская научно-техническая конференция «Нейроинформатика-2010»: Сборник научных трудов. Ч.2. М.: НИЯУ МИФИ, 2010, С. 326-334

ПРИМЕНЕНИЕ COMSOL MULTIPHYSICS ДЛЯ МОДЕЛИРОВАНИЯ RFID СИСТЕМ

Жуйков В.В.

Научный руководитель: Руководитель СКБ «Наносистемы» Елсуков К. А.

МГТУ им. Н.Э. Баумана, Москва, Россия

APPLICATION COMSOL MULTIPHYSICS FOR MODELLING RFID OF SYSTEMS

Zhuikov V.V.

Research supervisor: Supervisor of SDO «Nanosystems», Yelsukov K. A.

BMSTU, Moscow, Russia

Аннотация

В последние годы RFID-системы стали очень популярными во многих сферах услуг, продажи и распределения, в промышленности, производстве и системах, связанных с потоком данных. Однако проектирование и разработка RFID-систем является достаточно сложной задачей и требует больших материальных затрат на анализ и получение экспериментальных данных. Применение физического моделирования переводит проектирование и расчёт сложных RFID-структур на новый уровень. В данной работе представлено моделирование физической модели RFID-системы и проанализированы её свойства с помощью пакета программ COMSOL Multiphysics.

Abstract

The last years RFID-systems became very popular in many spheres of services, sales and allocations, in the industry, production and the systems linked to the data flow. However designing and development of RFID-systems is enough challenge and demands the big material inputs on the analysis and reception of experimental data. Application of physical modelling make designing and calculation of difficult RFID-structures to new level. In the given operation modelling of physical model of RFID-system is presented and its properties by means of software package COMSOL Multiphysics are parsed.

Введение

В настоящее время радиочастотная идентификация (RFID – англ. *Radio Frequency Identification*) является одной из быстро развивающихся технологий в области беспроводных коммуникаций ближнего действия. Ключевыми элементами в любой системе RFID являются опрашивающее приемо-передающее устройство, или ридер (interrogator, reader) и радиометка, транспондер или тэг (tag). Тэг состоит из небольшой планарной антенны, с которой одним из способов интегрируется микрочип, изготовленный с использованием полупроводниковой технологии [2].

Области применения данной технологии обширны: она используется, чтобы идентифицировать, проследить, рассортировать и обнаружить неограниченное количество предметов, включая людей, документы, транспортные средства, одежду, контейнеры, и т.д. Она может быть использована для автоматизации производственных процессов, автоматизации систем управления, организация систем контроля доступа и систем безопасности, контроля и учёт рабочего времени, построение дисконтных и логических систем, защита товаров и документов от подделок.

RFID – бесконтактная технология, носителем информации которой является радиоволна. Для обеспечения работы системы не требуется ни контакта со считывателем, ни прямой видимости считывателя (в отличие от систем с использованием штрих-кодирования, магнитных и smart карт). Надежная работа гарантирована при работе на агрессивных средах и неблагоприятных климатических условиях.

1. Основные преимущества технологии RFID

- для RFID не нужен контакт или прямая видимость;
- за единицу времени может идти работа с большим количеством меток;
- метки могут нести большое количество информации;
- метки читаются быстро и точно (приближаясь к 100%);
- информация на метке может быть зашифрована;
- возможность построения интеллектуальных, географически-разрозненных систем;

- RFID может использоваться в агрессивных средах, а метки могут читаться через краску, грязь, пар, воду, пластмассу, древесину;
- пассивные RFID-метки имеют фактически неограниченный срок эксплуатации;
- RFID-метки несут большое количество информации;
- RFID-метки практически невозможно подделать.

2. Принцип работы RFID-систем

Система RFID состоит из двух основных частей (рис.1) [1]:

1. Тега (tag) или транспондера (transponder) с антенной на печатной плате;
2. Ридера (reader) с большой RF-антенной.



Рис.1. Принцип работы RFID-систем.

Пассивные системы радиочастотной идентификации (RFID) относятся к классу беспроводных технологий передачи и сбора данных, в которых опрашивающее приемопередающее устройство, или ридер (reader, interrogator) излучает модулированный RF сигнал в направлении радиометки, или тэга (tag, transponder), который состоит из антенны и интегрированного с ней полупроводникового чипа [3]. Необходимую для активации мощность чип получает через антенну из падающей электромагнитной волны. Активированный чип изменяет свой входной импеданс в соответствии с заложенной в его память информацией и тем самым модулирует отраженный ответный сигнал, детектируемый приемным устройством ридера.

3. Моделирование опрашивающего приемопередающего устройства, или ридера (interrogator, reader) и радиометки, транспондера или тэга (tag)

Взаимодействие антенн носит индуктивный характер и характеризуется коэффициентом взаимной индукции L_{12} . Взаимная индукция – это явление возникновения ЭДС электромагнитной индукции в одном из контуров при изменениях силы тока в другом или при изменении взаимного расположения этих двух цепей (контуров).

Transponder-антенна, как правило, выполняется на керамической подложке [4]:

Распространённый тип reader-антенны выполняется в виде больших двух катушек [4]:

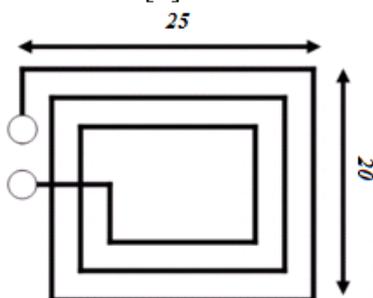


Рис.2. Модель transponder-антенны.

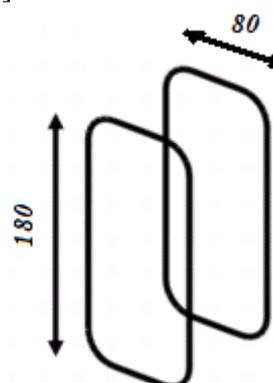


Рис.3. Модель reader-антенны.

Коэффициент взаимной индукции определяется как отношение общего магнитного потока, проходящего через одну reader-антенну, к единице тока, протекающего в другой transponder-антенне [5]:

$$L_{12} = \frac{\iint_{S_2} \mathbf{B} \cdot \mathbf{n} dS}{I_1}, \quad (1)$$

где S_2 – площадь reader-антенны,

\mathbf{B} – вектор магнитной индукции reader-антенны,

I_1 – ток, протекающий в transponder-антенне;

$\mathbf{n} = (n_x, n_y, n_z)$ – вектор нормали к поверхности reader-антенны;

dS – бесконечно-малая площадь transponder-антенны.

Преобразуем эту формулу, используя вектор напряжённости магнитного поля [5]:

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (2)$$

Используя теорему Стокса, получаем [5]:

$$L_{12} = \frac{\iint_{S_2} (\nabla \times \mathbf{A}) \cdot \mathbf{n} dS}{I_1} = \frac{\oint_{\Gamma_s} \mathbf{A} \cdot \mathbf{t} dl}{I_1}, \quad (3)$$

где $\mathbf{t} = (t_x, t_y, t_z)$ – единичный вектор, касательный к кривой Γ_s , dl – бесконечно-малая длина transponder-антенны.

Поскольку связь преобладает в ближнем поле действия индуктивных эффектов, достаточно вычислить взаимную индуктивность для статического случая (частота равняется нулю), пренебрегая ёмкостными эффектами наряду с волновыми явлениями распространения волн.

Для моделирования RFID необходимо применить модуль [4]: 3D-Magnetostatics, находящийся в AC/DC модуле пакета COMSOL Multiphysics. Вектор напряжённости магнитного поля для RFID системы имеет вид:

$$\mathbf{A} = (A_x, A_y, A_z).$$

Этапы физического моделирования состоят из 5 основных частей:

- 1) Создание геометрии;
- 2) Подключение физики;
- 3) Граничные условия;
- 4) Создание сетки;
- 5) Создание модели.

1) Создание геометрии

а) Создание геометрии transponder - антенны в 2D

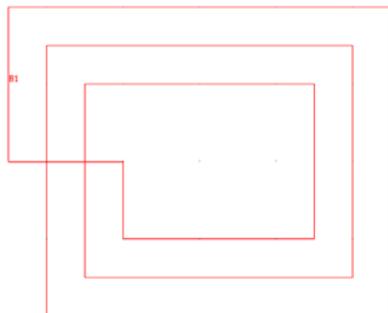


Рис.4. Transponder-антенна.

б) Создание геометрии reader - антенны в 2D

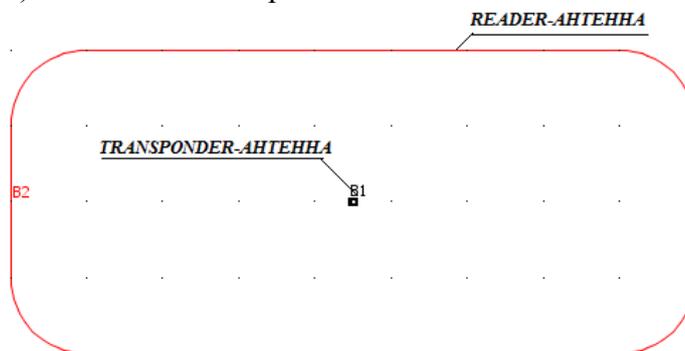


Рис.5. Reader-антенна.

в) Экспорт в 3D пространство

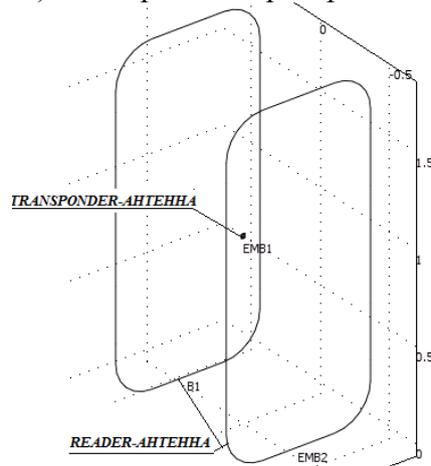


Рис.6. Взаимное расположение антенн.

Необходимо ограничить расчётную область, так как конечный расчёт требует конечных размеров сетки. Для этого построим сферу диаметром 3 метра с центром в середине системы антенн.

2) Подключение физики

Параметр материала, который можно изменить для моделирования магнитостатики – относительная магнитная проницаемость. Однако предположим, что сфера наполнена воздухом, поэтому $\mu_r=1$ [6].

3) Определение граничных условий

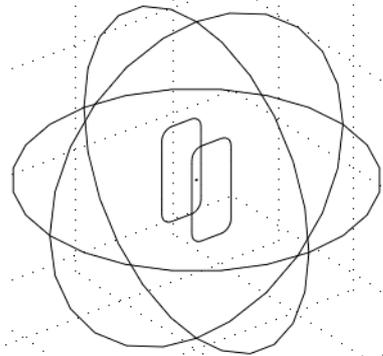


Рис.7. Каркас сферы.

а) Примем, что поверхность сферы изолирована от магнитных полей. Действительно, тангенциальная составляющая вектора магнитной индукции стремится к нулю от антенн к бесконечности.

б) Определим знак величины протекающего тока reader-антенны.

Для определения знака величины тока применим правило правой руки для граней: Если направление тока – от оси Z к X и большой палец указывает на ось Y, то такое направление считается положительным, иначе – отрицательным.

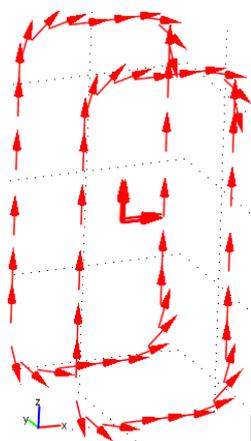


Рис.8. Направление тока в reader-антенне.

4) Построение сетки (16000 степеней свободы)

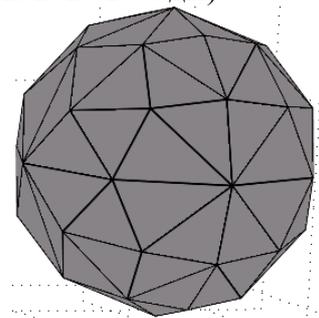


Рис.9. Сетка.

5) Создание модели

а) Значение и силовые линии магнитного потока

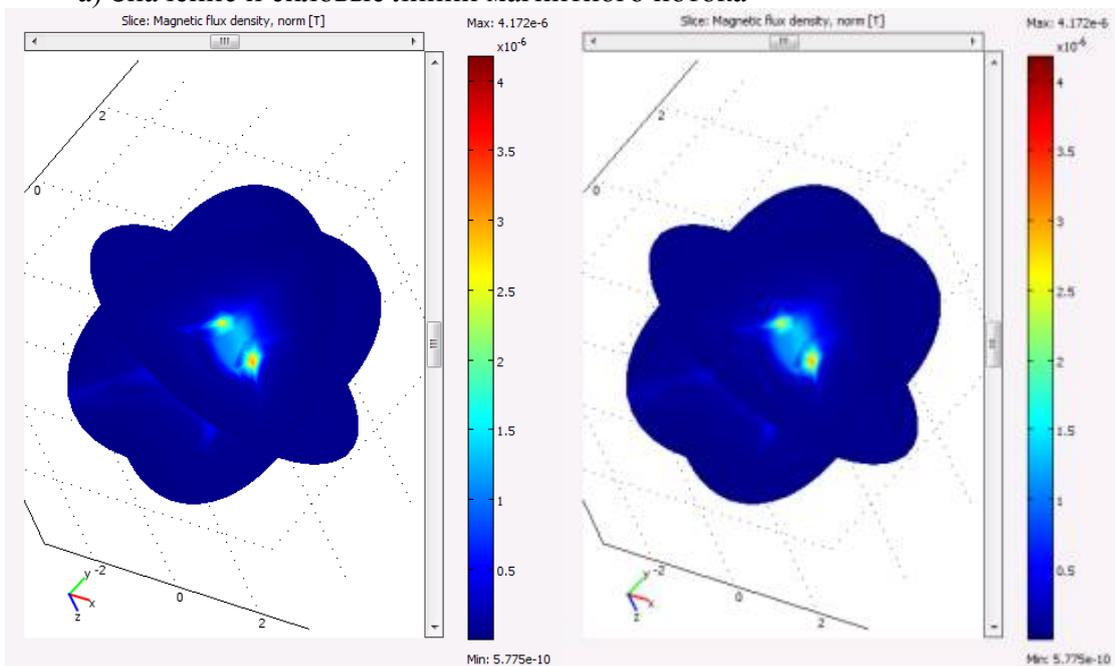


Рис.10. Распределение плотности напряжённости магнитного поля.

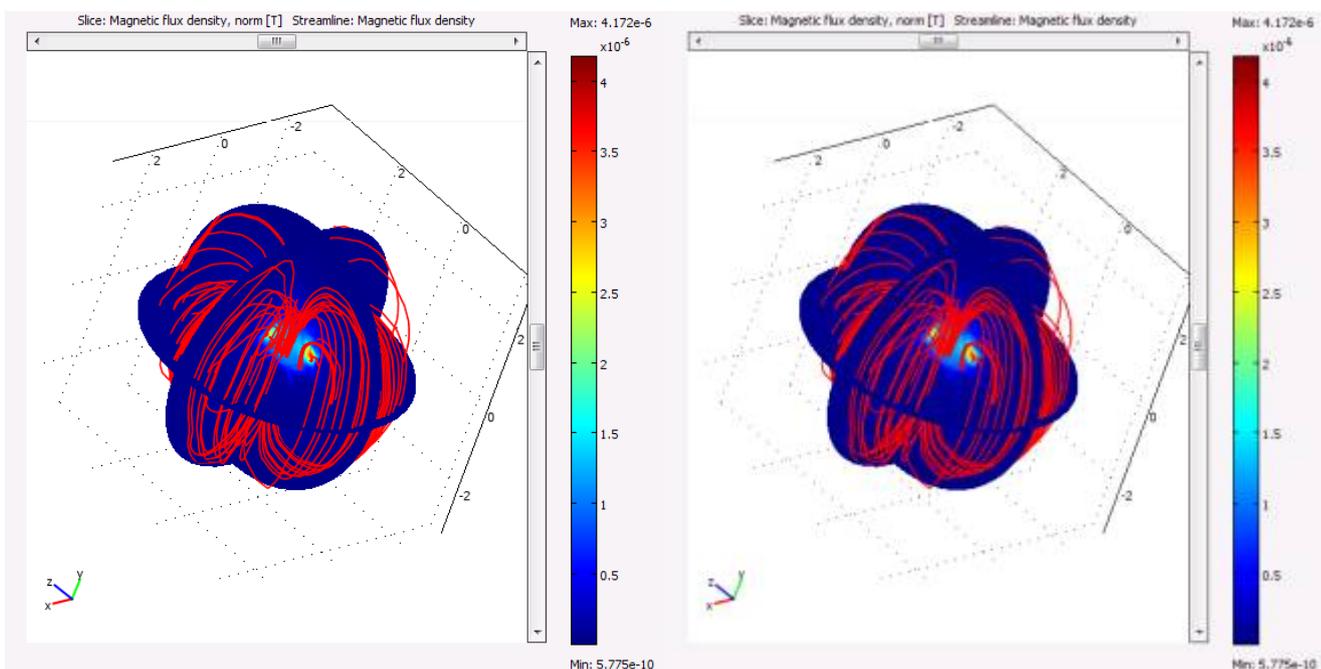
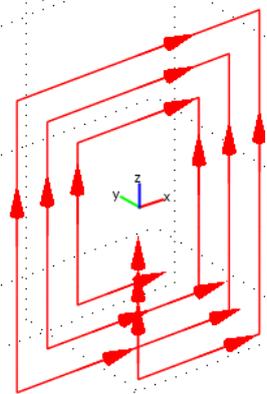


Рис.11. Силовые линии магнитного потока.

Силловые линии магнитного поля модели удовлетворяют ожидаемому результату. Из рисунка видно, что максимальная плотность напряжённости магнитного поля – в окрестности reader-антенны.

б) Вычисление взаимной индуктивности



Для определения знака величины тока применим правило правой руки для граней:

Если направление тока – от оси Z к X и большой палец указывает на ось Y, то такое направление считается положительным, иначе – отрицательным.

Рис.12. Направление тока в transponder-антенне.

Вычисляем значение взаимной индуктивности [5]:

$$L_{12} = \frac{\oint_{\Gamma_2} \mathbf{A} \cdot d\mathbf{l}}{I_1} = \oint_{\Gamma_2} \mathbf{A} \cdot \mathbf{t} dl \quad (4)$$

$$\mathbf{A} = A_x \cdot \mathbf{n}_x + A_y \cdot \mathbf{n}_y + A_z \cdot \mathbf{n}_z, \quad (5)$$

где A_x, A_y, A_z – проекции вектора напряжённости магнитного поля на направления OX, OY, OZ. $\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z$ – единичные векторы, направленные вдоль осей X, Y, Z.

В результате получим значение:

Value: 9.653366e-10, Expression: L12, Phase: 0 degrees

Value: 9.653366e-10, Expression: L12, Phase: 0 degrees

(6)

Значение взаимной индукции: $L_{12} = 9,653366 \cdot 10^{-10} \approx 1 \text{ нГн}$.

в) Приведём решение с более тонкой настройкой сетки (60000 степеней свободы):

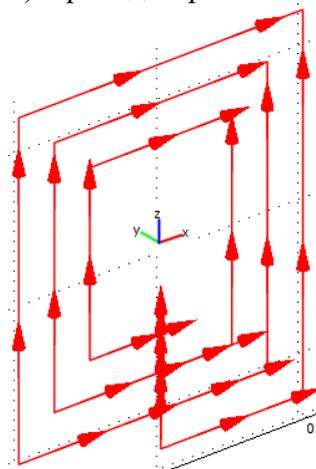


Рис.13. Направление тока в transponder-антенне с более тонкой настройкой сетки.

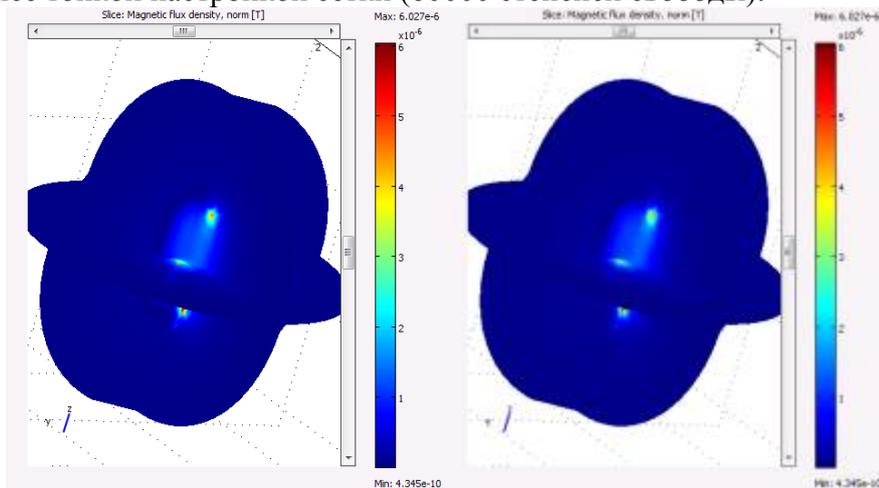


Рис.14. Модель распределения плотности напряжённости магнитного поля с более тонкой настройкой сетки.

г) Оптимизируем сетку ещё лучше. Сетка должна точно описывать форму антенны. В результате получается 180000 степеней свободы. Для большей наглядности построим плотность распределения магнитного потока в логарифмическом масштабе.

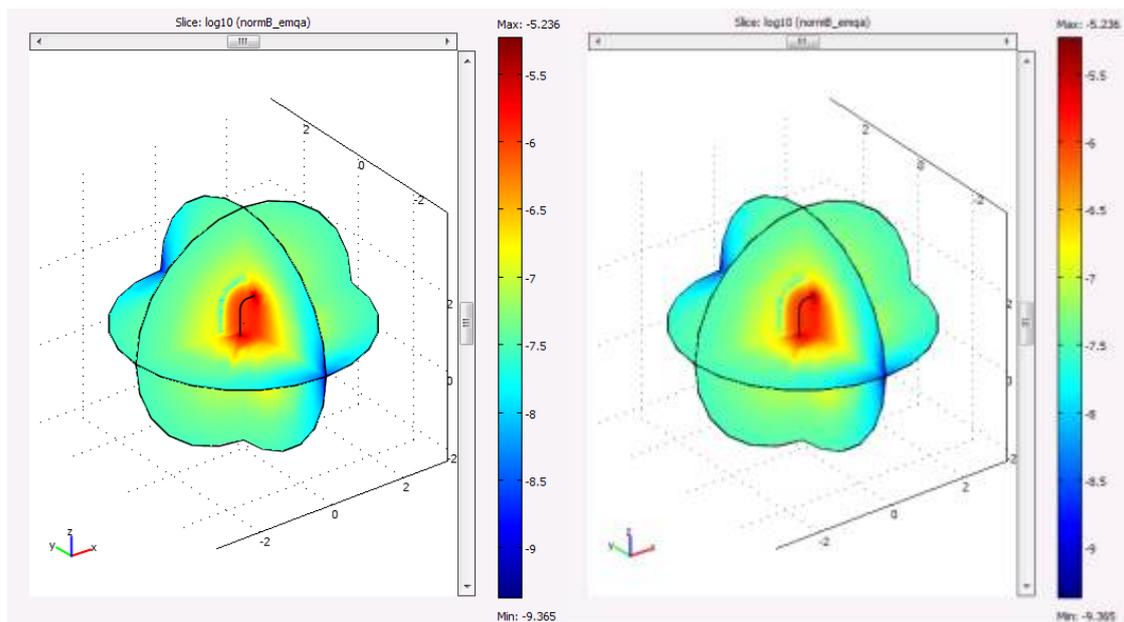


Рис.15. Плотность распределения магнитного потока в логарифмическом масштабе. Значение коэффициента взаимной индукции в этом случае будет равно:

Value: 9.848516e-10, Expression: L12, Phase: 0 degrees

Value: 9.848516e-10, Expression: L12, Phase: 0 degrees

(7)

Значение взаимной индукции: $L_{12} = 9,848516 \cdot 10^{-10} \text{ Гн} \approx 1 \text{ нГн}$

д) Проверим правильность выбора сетки для reader-антенны:

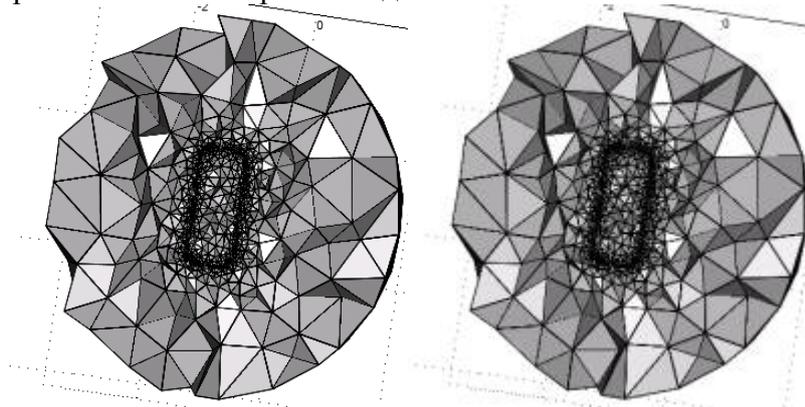


Рис.16. Разрез сферы.

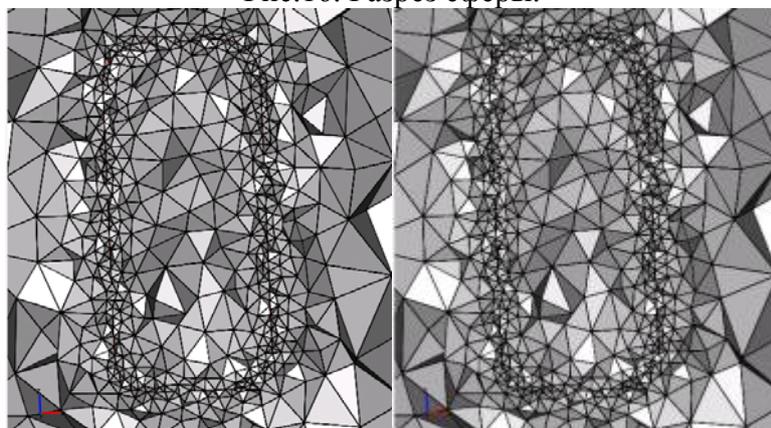


Рис.17. Сетка для reader-антенны.

Видно, что сетка для reader-антенны подобрана достаточно точная, поэтому вычисления можно считать с большой вероятностью правильными.

е) Визуализация линий магнитной индукции, построение модели RFID-системы.

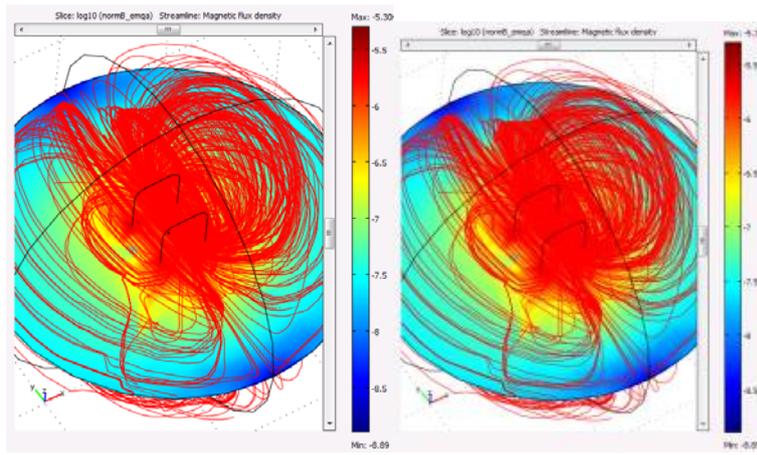


Рис.18. Модель RFID-системы (вид 1).

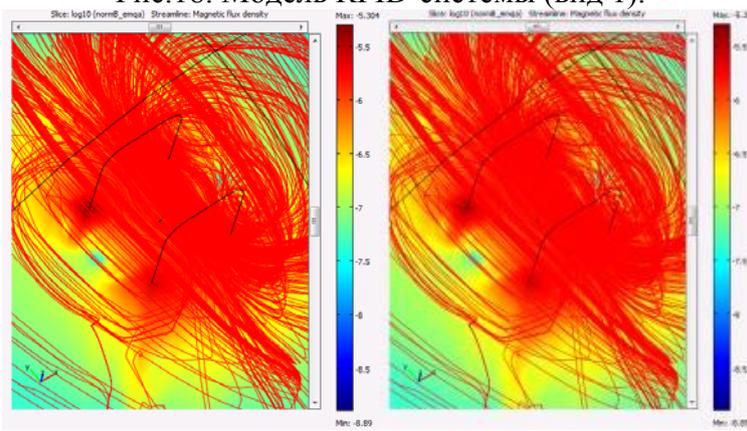


Рис.19. Линии магнитной индукции.

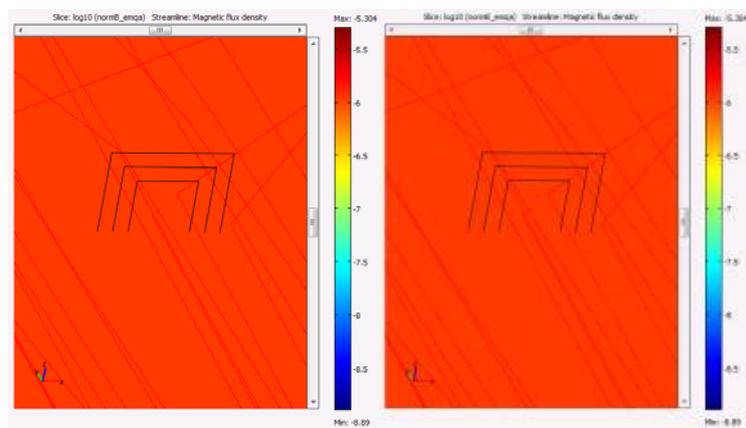


Рис.20. Transponder-антенна в магнитном поле.

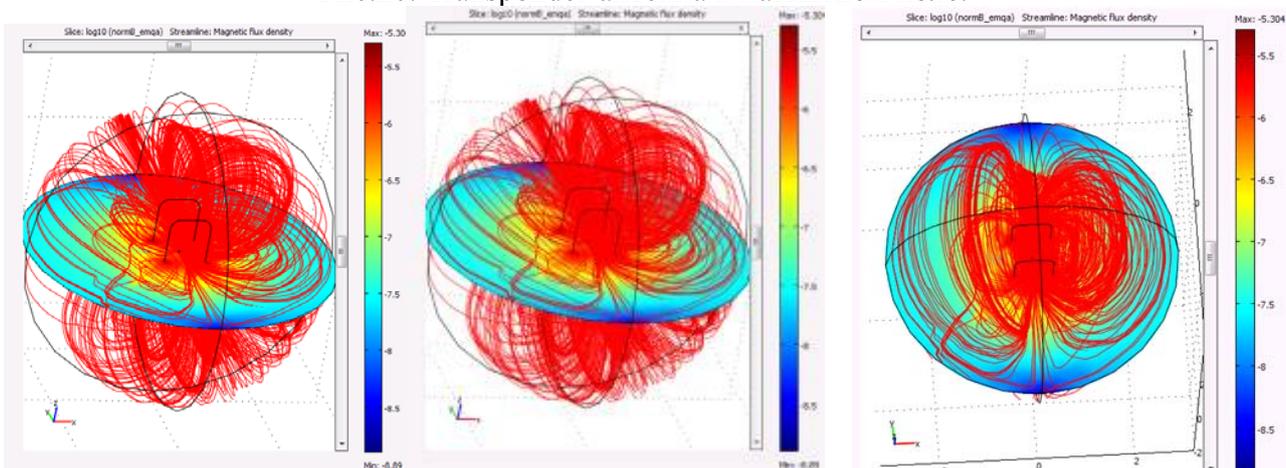


Рис.21. Модель RFID-системы (вид 2).

Рис.22. Физическая модель RFID-системы.

В результате моделирования получили качественное представление о распространении линий магнитной индукции и плотности распределения магнитного потока.

ж) Моделирование при гармоническом воздействии

Модель RFID тега также может быть решена при гармоническом воздействии, то есть при определённой частоте переменного тока (в отличие от случая постоянных электрических токов и полей).

При низких частотах значение взаимной индукции не сильно отличается от значения при постоянном значении тока.

Увеличение частоты тока RFID-системы увеличивает потери на излучение, поэтому необходимо учитывать данный эффект, и добавить поглощающие слои или дополнительные граничные условия за пределами сферы[6].

Выводы

Вычисленное значение магнитной индукции L_{12} составляет около 1нГн.

На модели (рис.22.) представлены:

- а) силовые линии магнитного потока; б) шкала десятичного логарифма интенсивности магнитного потока;
- в) плотность распределения магнитного потока.

Антенны, работающие в системах RFID, имеют разнообразную конструкцию, размеры и требуемые диаграммы направленности. Причем их размеры имеют тенденцию к уменьшению. Тэги становятся все более компактными. Поэтому проектирование тэгов выделяется в класс самостоятельных задач, тесно связанной с задачей согласования комплексных импедансов[6]. Проектирование согласующих цепей (reader-transponder) в конструкциях антенн является актуальной задачей.

Применение Comsol Multiphysics переводит проектирование сложных структур на новый уровень. В результате электромагнитного анализа имеется возможность рассчитать согласующие структуры произвольной формы, проанализировать силовые линии и плотность распределения напряжённости магнитного поля. Результаты проектирования, с использованием программы становятся не менее точными, чем экспериментальное измерение.

Анализ конструкции пары «чтения-передачи» (reader-transponder) выполняется с помощью данной программы очень эффективно, при этом мощные современные компьютеры позволяют моделировать полностью пространство от антенны ридера до антенны тэга даже при переменном гармоническом воздействии.

Литература

1. Андренко А. С., Бабаскин А. А., Калиничев В. И., Курушин А. А. Согласование антенны тэга с микрочипом для систем радиочастотной идентификации //М.: Журнал радиоэлектроники, №12, 2007 - <http://jre.cplire.ru/jre/dec07/2/text.html>.
2. Багдасарян С. А., Гуляев Ю. В. Радиочастотная идентификация с использованием технологии ПАВ. // М.: Наука и технология в промышленности, 2005. № 1. сс. 54-60.
3. Финкенцеллер К. Справочник по RFID. Теоретические основы и практическое применение индуктивных радиоустройств, транспондеров и бесконтактных чип-карт // М.: Додека, 2008. – 496 с.
4. Физическое моделирование в среде COMSOL (<http://www.comsol.com>).
5. *Balanis K.* Antenna Theory: Analysis and Design. 2nd Edition: John Wiley and Sons, 2001.
6. *Nikitin P. V. and Rao K. V. S.* Performance of RFID Tags with Multiple RF Ports // Proc. Int. Symp. IEEE AP. June 2007, Honolulu, HI, USA. PP. 5459-5462.

АНАЛИЗ МЕТОДОВ И ТЕХНОЛОГИЙ РАСПОЗНАВАНИЯ 3D ОБЪЕКТОВ

Баранов А.В.

Научный руководитель доц., к.т.н. Волосатова Т.М.
МГТУ имени Н.Э.Баумана, РК-6 САПР, Москва, Россия

THE ANALYSIS OF METHODS AND TECHNOLOGIES OF RECOGNITION 3D OBJECTS

Baranov A.V.

The supervisor of studies prof. Volosatova T.M.
RK-6 CAD Systems, Bauman Moscow State Technical University, Moscow, Russia

Аннотация

Компьютерная графика и анимация стали самыми популярными и неотъемлемыми атрибутами современных кинофильмов, рекламных роликов и видеоигр. В наши дни технология распознавания 3D предлагает нам сканирование объектов, а так же захват их движений. 3D камеры представляют собой сканер с широкопольными 3D и мегапиксельными 2D сенсорами, технология позволяет регистрировать форму и текстуру поверхности объекта в режиме фото- или видеосъемки. Это свойство камер предоставляет замечательные возможности для съемки движений всего человеческого тела или иных движущихся объектов в режиме реального времени. Все данные получаются путём проецирования на объекты сцены специальной решетки под параллаксным углом. Искажения проекции решетки, созданные геометрией объектов, позволяют рассчитать точное положение каждой её точки в трёхмерном пространстве. Полученные точки триангулируются, образуя полигональную поверхность.

Abstract

Computer graphics and animation have made a significant impact on entertainment and multimedia industries and have, therefore, become the most popular and essential features in movies, commercials and video games. In our days technology of 3D recognition provides us object scanning and motion capture. 3D scanners use equipped with wide-field-of-view 3D and mega-pixel 2D sensors, the technology allows capturing both shape and texture of the object surface in a snap-shot or video mode. This feature provides great opportunities for capturing movements of the entire human body or other motion objects in a real-time mode. The data is acquired by projecting a special light pattern, the grid, onto the scene objects at a parallax angle. Grid distortion produced by geometry of objects is used to calculate the exact 3D coordinates of each point. These points form the basis for the triangulated.

Современный этап развития компьютерной графики и анимация характеризуется следующей общей тенденцией. Она заключается в том, что объектом их обработки являются различного рода изображения, достаточно сложные по структуре и с неоднородными статистическими свойствами. Задачей таких систем является не только такая пассивная функция, как формирование самих изображений, но, в первую очередь, понимание изображений – распознавание и оценка их параметров.

Трёхмерное сканирование – современная перспективная технология, позволяющая на более высоком уровне решать задачи, в которых приходится работать с объектами сложной пространственной формы. Такие задачи возникают на производстве, в дизайне, медицине, биометрии и т.д.

Идентификация личности по трёхмерной модели лица (Рисунок 1) является одной из наиболее перспективных и сложных направлений. Такие системы возможно использовать для повышения надежности систем контроля доступом в сочетании с любыми исполнительными устройствами: электромеханическими замками, турникетам, раздвижными дверьми и т.п. Использование устройства почти полностью исключает влияние человеческого фактора при проверке документов, например, усталость охранника при большом потоке людей.



Рис.1. Реконструированная модель лица

В наши дни технология распознавания 3D предлагает нам сканирование объектов, а так же захват их движений. Сканирование объектов осуществляется по двум основным алгоритмам.

Первый включает в себя условие наличия подсветки со структурированным светом. Все данные получают путём проецирования на объекты сцены специальной решетки под параллаксным углом. Искажения проекции решетки, созданные геометрией объектов, позволяют рассчитать точное положение каждой её точки в трёхмерном пространстве. Полученные точки триангулируются, образуя полигональную поверхность. Данная технология предполагает:

- подсвечивание объекта, 3D модель которого необходимо реконструировать, специальными шаблонами подсветки (Рисунок 2);
- идентификацию характерных особенностей шаблонов подсветки на цифровых изображениях подсвеченного объекта;
- применение известного метода оптической триангуляции для реконструирования 3D точек поверхности объекта, основываясь на результатах проведенной идентификации

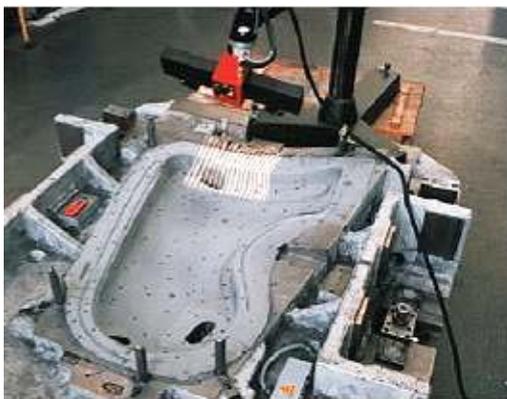


Рис.2. Использование структурированной подсветки при 3D сканировании.

Особенностью второго алгоритма является отсутствие какого либо лазерного или другого дополнительного света, направленного на объект (структурированной подсветки). Использование нескольких камер позволяет добиться отличных результатов при реконструкции объектов. Например, оптимальным является использование трех синхронизированных во времени камер, разнесенных друг относительно друга на расстояние порядка 20 см между камер высокого разрешения. Такой режим улучшает среднестатистические показатели реконструированной поверхности. 3D реконструкция модели осуществляется поэтапно:

- построение поля 3D точек;
- предварительная фильтрация точек;
- построение и окончательной фильтрация триангулированной модели;
- совмещение нескольких поверхностей, полученных в разные моменты времени (видеоряд), в одну общую объединенную 3D модель(Рисунок 3).

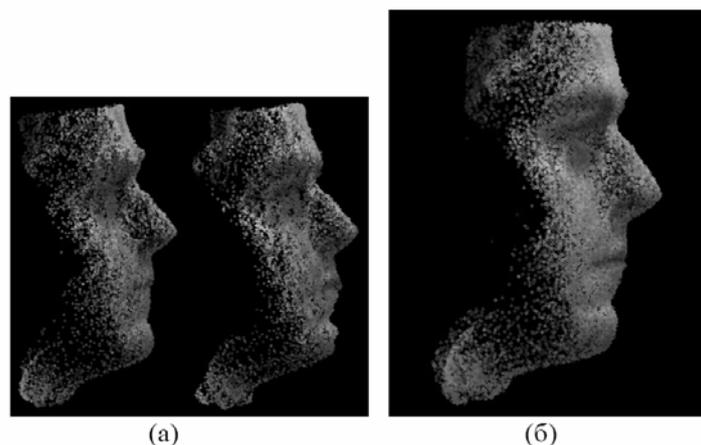


Рисунок 3. Обработка 3D видео. (а) – 3D модели лица в разные моменты времени, (б) – после совмещения 3D моделей в общую модель.

Устройства на базе технологий 3D-сканирования могут быть интегрированы в сложные пространственные системы для съемки специфических объектов. Благодаря высокой точности и скорости съемки технология может использоваться в ручных и мобильных системах.

Задачи распознавания плоских, 2D изображений отходят на второй план, оставляя за собой широко используемые в системах безопасности технологии распознавания номерных знаков автомобилей, распознавания движущихся объектов с возможностями трассировки и ведения целей. Как показывает обзор достигнутых результатов в области обработки пространственных 3D объектов технология сканирования показывает лучшие результаты, по сравнению обработкой 2D по критериям надежности, точности. В настоящее время вышеуказанные методы уже позволяют снимать в режиме реального времени видео в формате 3D с частотой 15 кадров в секунду. Однако стоимость оборудования, несмотря на развитие отрасли, оставляет желать лучшего.

Основными векторами для дальнейшего развития систем распознавания станут: биометрия, распознавание и сканирование движущихся объектов, охранные системы, применения в сканирования на производстве для контроля качества.

Литература

1. Face Recognition for Smart Interactions Ekenel, H.K.; Stallkamp, J.; Gao, H.; Fischer, M.; Stiefelhagen, R. Multimedia and Expo, 2007 IEEE International Conference on Volume , Issue , 2-5 July 2007 Page(s):1007 – 1010
2. Wei Jiang; Jian Lu, Panoramic 3D Reconstruction by Fusing Color Intensity and Laser Range Data Robotics and Biomimetics, IEEE International Conference on Volume , Issue , 17-20 Dec. 2006 Page(s):947 – 953
3. Real-time face recognition for human-robot interaction Cruz, C.; Sucar, L.E.; Morales, E.F Automatic Face & Gesture Recognition, 2008. FG apos;08. 8th IEEE International Conference on Volume , Issue , 17-19 Sept. 2008 Page(s):1 - 6
4. Коробейников А. П. Методы распознавания образов: Учеб. пособие. Ростов-на-Дону: Издательский центр ДГГУ, 1999. 51с
5. De Wansa Wickramarante V.K., Ryazanov V.V., Vinogradov A.P. Accurate Reconstruction of 3D Model of a Human Face Using Structured Light //Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications, Vol. 18, No.3 – М.: 2008. с. 442-446.
6. Artec Group. Компьютерная графика и анимация, 2010 <http://artec-group.ru/3dscanning/cg.html>
7. Визильтер Ю.В. Желтов С.Ю. Князь В.А. Обработка и анализ цифровых изображений с примерами на LabVIEW и IMAQ Vision, - М. 2007 с. 464

СИНТЕЗ ИЗОБРАЖЕНИЙ ИЗ ФРАГМЕНТОВ ВИДЕОИЗОБРАЖЕНИЙ В СИСТЕМАХ МОНИТОРИНГА И РАЗВЕДКИ

Горин Я.А.

Научный руководитель к.т.н, доцент. Волосатова Т.М.

Кафедра РК6, МГТУ им. Н.Э. Баумана, Москва, Россия

SYNTHESIS OF IMAGES FROM THE VIDEOIMAGE FRAGMENTS IN MONITORING AND EXPLORATION SYSTEMS

Gorin J.A.

The scientific chief: Ph.D. senior lecturer Volosatova T.M.

Department RK6, BMSTU, Moscow, Russia

Аннотация

В работе приведен обзор задачи синтеза изображений из фрагментов видеоизображений. Рассмотрены основные особенности, налагаемые применением синтеза в системах мониторинга и разведки. Разработано ПО для синтеза изображения по видеофрагментам. Проиллюстрированы результаты тестов, имитирующих реальные условия, в которых программа может применяться.

Abstract

An overview of the problem of synthesizing images from the fragments of videoimages is given. Main features imposed by the use of synthesis in the systems of monitoring and exploration are considered. Software for the synthesis of images from video is developed. The results of tests that simulate actual conditions in which the program can be applied is illustrated in this work.

Под синтезом изображения из фрагментов понимают процесс, обратный выделению подизображений. Из нескольких изображений, имеющих общую часть, синтезируется суммарное изображение, представляющее наложение исходных изображений так, что бы общая часть совпала. При этом «общая часть» может немного отличаться. Основная задача синтеза изображений: нахождение относительного смещения изображений.

Для систем мониторинга встает вопрос быстродействия при выполнении операций синтеза. Фотограф может позволить себе подождать и час, и день, пока компьютер обрабатывает его фотографии, но если система мониторинга и разведки ЧС будет обрабатывать полученные снимки (которые и так поступают с довольно большой задержкой) слишком долго, то об оперативном реагировании не может быть и речи. Если же произойдет переход системы мониторинга и разведки ЧС с космических средств наблюдения (спутников) на воздушные (дирижабли), то объем информации, которую необходимо обрабатывать, может сильно возрасти.

Исследовав все требования, решено оформить программный продукт в виде статической библиотеки, поставляемой вместе с набором интерфейсных файлов. Для обеспечения расширяемости надо использовать объектно-ориентированный подход и разбиение программы на независимые модули. Для обеспечения кросс-платформенности не следует использовать платформу-зависимый API и библиотеки.

Синтез изображения из фрагментов видеоизображений

Синтез изображения из фрагментов видеоизображений можно рассматривать как последовательный синтез изображения из отдельных кадров. На каждой итерации синтезируется изображение из «текущего» изображения (изначально – первый кадр видеофрагмента) и первого необработанного кадра видеофрагмента. Синтезированный кадр становится текущим. Итерации продолжаются до окончания кадров в видеофрагменте. Получившийся итеративный процесс имеет следующие особенности:

3. Небольшое относительное смещение обрабатываемых изображений.
4. Большое количество информации, которую необходимо обработать.

Для аэрофильмов так же характерны следующие свойства:

Постоянное (или плавно изменяющееся) направление вектора смещения изображения.

С учетом этих особенностей возможны несколько режимов обработки видеофрагмента:

- Обрабатывается каждый кадр – повышается точность и время вычислений. Может применяться при несрочной обработке видеофрагментов, полученных ранее, с целью дальнейшей обработки полученного изображения.
- Обрабатывается каждый второй (третий, четвертый, ...) кадр – понижается точность и время вычислений. Может применяться при обработке видеоизображения «на месте».

Одной из проблем при синтезе изображения из фрагмента видеоизображения – изменение ориентации камеры. Т.к. предварительная ручная обработка каждого кадра невозможна, то необходима система автоматической предварительной обработки кадров, которая должна уметь:

- Вращать кадр, обеспечивая соориентированность его с текущим изображением.
- Регулировать яркость изображения.
- Обрезать кадр, в случае наличия краевых эффектов (например, черной полосы по краю кадра).

Выделение функциональных подсистем программного обеспечения

Перед разрабатываемым программным обеспечением стоит задача синтеза изображения из фрагментов видеоизображения.

Основная функциональность программного обеспечения – именно синтез изображения из фрагментов, однако, машинное представление изображения – это набор пикселей, каждый из которых задается несколькими параметрами. Таким образом встает задача первичной обработки видео, с целью преобразования его в набор кадров, пригодных для дальнейшей обработки.

Полученное изображение так же необходимо хранить, т.к. полученное изображение может обрабатываться следующими системами не сразу, а по мере необходимости.

Процесс обработки необходимо логировать, что бы в случае сбоев было проще обнаружить, локализовать и исправить проблему.

Таким образом, можно выделить следующие функциональные подсистемы:

- 5) Подсистема обработки изображений (синтез изображения из отдельных фрагментов).
- 6) Подсистема обработки видеофайлов (считывание, декодирование и разбиение на кадры).
- 7) Подсистема обработки файлов изображений (запись изображения в файл).
- 8) Подсистема логирования.

Структура программы

Библиотека предоставляет набор интерфейсов и их реализаций. Пользователь может разрабатывать собственные реализации предоставленных интерфейсов и передавать их в библиотеку (при условии, что они удовлетворяют требованиям библиотеки). Также можно использовать библиотечные реализации в пользовательских, таким образом расширяя их без повторного написания кода.

Описание работы программы

Целью данной программы является предоставление пользователю возможности “сшивать” видеофайлы в формате MPEG-4. В результате работы программы получается изображение в формате TGA. Программа предназначена для синтеза изображений из фрагментов видеоизображений аэрофильмов. В данном конкретном случае в качестве исходных экспериментальных данных для синтеза изображений использовались три видеозаписи, полученные с помощью IP-камер AXIS 213 и SONY RZ25P.

Литература

1. Мосягин Г. М., Немтинов В. Б. Преобразование сигналов в оптико-электронных приборах систем управления летательных аппаратов. – М.: Машиностроение, 1980 – стр. 182
2. Л.П. Лазарев, В. Я. Колючкин, А. Н. Метелкин и др. Автоматизация проектирования оптико-электронных приборов: Учебное пособие для оптических специальностей вузов – М.: Машиностроение, 1986 – стр. 216
3. Гриффитс А. GCC. Полное руководство. – ДиаСофт, 2004.
4. GNU Make- <http://www.linux.org.ru/books/GNU/Gmake.htm>
5. Преобразование Фурье. – <http://ru.wikipedia.org>
6. Преобразование Фурье и его свойства. – <http://alglib.sources.ru>
7. БПФ комплексной функции (прямое и обратное). – <http://alglib.sources.ru>
8. Описание видео форматов. – <http://ru.wikipedia.org>