

Рис. 2 – Экранная форма разработанной программы

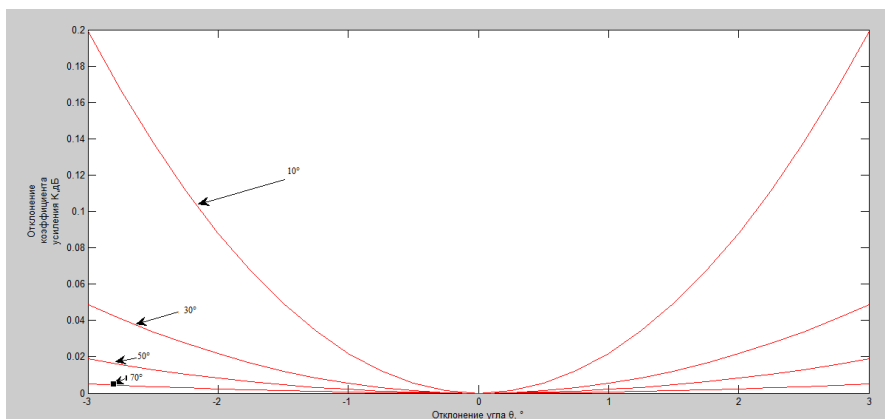


Рис. 3 – График зависимости отклонения коэффициента усиления от отклонения угла приема полезного сигнала при различных углах между лучом приема полезного сигнала и углом приема помехи (антенна с двумя элементами)



Исходя из полученных графиков (см. рис. 3), можно получить следующие выводы:

1. Чем выше разность углов между принимаемыми полезным и помеховыми сигналами, тем выше погрешность усиления, при ошибке определения направления на источник сигнала. При данном угле прихода полезного сигнала, ослабление сигнала может достигать 0,2 дБ, хотя при данной оценке не участвует влияние со стороны помехового сигнала, так как он не был подавлен полностью.
2. Чем меньше ошибка в определении направления на источник сигнала, тем более точно можно получить коэффициент усиления.

Литература

1. Баланис К.А. Введение в смарт-антенны. [Текст] / К. А. Баланис, П.И. Ионидес. – М.: Техносфера, 2012. – 200 с.

М.А. Посыпкин

ОТКРЫТАЯ ПРОГРАММНАЯ АРХИТЕКТУРА ДЛЯ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ МЕТОДОВ ОПТИМИЗАЦИИ

(Вычислительный центр им. А.А. Дородницына
Федерального исследовательского центра «Информатика и управление» Российской академии наук)

В последнее время значительный интерес и внимание ученых вызывает детерминированная глобальная оптимизация, в рамках которой удается не только получить решение, но и численно доказать его точность [1]. Такие алгоритмы обычно очень трудоемки в вычислительном отношении. Одним из способов решения проблемы высокой ресурсоемкости алгоритмов, гарантирующих оптимальность решения, является применение методов параллельных и распределенных вычислений [2].

Вычислительная схема ветвей и границ является основой для реализации многих методов оптимизации, в частности, метода неравномерных покрытий [1,3], методов ветвей и отсечений [3], многих методов дискретной оптимизации. Отличаясь в деталях, варианты МВГ для различных задач имеют общую базовую схему, которая состоит в декомпозиции исходной задачи на подзадачи с последующей обработкой подзадач и их «отсевом». Отсевом называется исключение подзадачи из дальнейшего рассмотрения по некоторым правилам. Наличие общей схемы дает возможность эффективно разделить проблемно-зависимую и проблемно-независимые части в программном комплексе.

В основу разрабатываемой программной архитектуры положены принципы расширяемости. Разделение проблемно-зависимой и независимой составляющих дает возможность легко добавлять новые алгоритмы на основе МВГ, оставляя неизменной параллельную часть. Разработан программный комплекс



для решения задач детерминированными методами, основанными на методе ветвей и границ (МВГ). Параллельная реализация ориентирована на параллельные системы с распределенной памятью и использует MPI [4].

Основной проблемой при реализации метода ветвей и границ на многопроцессорных системах является обеспечение равномерной загрузки процессоров. Для этого применяются методы балансировки нагрузки, которые перераспределяют эту нагрузку в процессе расчетов. В предлагаемом программном комплексе балансировщик выделен в отдельный модуль, а для описания управления балансировкой применяется формализм конечных автоматов. Это позволяет не только отделить детали реализации численного метода от логики распределения вычислительной нагрузки, но и открывает широкие перспективы по исследованию алгоритмов балансировки методами имитационного моделирования.

В рамках проекта по созданию открытой программной архитектуры для оптимизации на высокопроизводительных вычислительных системах был разработан симулятор параллельной системы [5], который позволяет выполнять алгоритм балансировки нагрузки, имитируя распределенную работу метода ветвей и границ, а также передачу данных между параллельными процессами. Для имитации МВГ используется случайный ветвящийся процесс с вероятностью ветвления, уменьшающейся пропорционально удалению от корня дерева. В основу моделирования положена концепция логических часов. Вместо реального решения подзадачи производится сдвиг логического таймера на заданную величину. Передача данных моделируется аналогично, при этом, время на принимающем процессе полагается равным временной метке полученного сообщения. Для облегчения изучения методов балансировки нагрузки разработана графическая среда [6], показывающая загрузку процессоров в разные моменты времени, визуализирующая обмены между ними. С помощью разработанной среды можно изучать как трассы реального выполнения, так и трассы, собранные при помощи симулятора.

Литература

1. Евтушенко Ю. Г. Численный метод поиска глобального экстремума функций (перебор на неравномерной сетке) // Журнал вычислительной математики и математической физики. – 1971. – Т. 11. – №. 6. – С. 1390-1403.
2. Стронгин Р. Г. и др. Параллельные вычисления в задачах глобальной оптимизации–М // М.: Издательство Московского университета. – 2013.
3. Евтушенко Ю. Г., Посыпкин М. А. Применение метода неравномерных покрытий для глобальной оптимизации частично целочисленных нелинейных задач // Журнал вычислительной математики и математической физики. – 2011. – Т. 51. – №. 8. – С. 1376-1389.
4. Snir M. MPI--the Complete Reference: The MPI core. – MIT press, 1998. – Т. 1.



5. Фомин А. Л. A software simulation tool for the parallel branch and bound method implementation // International Journal of Open Information Technologies. – 2015. – Т. 3. – №. 11. – С. 10-15.

6. Орлов Ю. В. Среда комплексного анализа производительности алгоритмов балансировки в параллельном методе ветвей и границ // International Journal of Open Information Technologies. – 2015. – Т. 3. – №. 9.

И.Д. Семенов, Е.И. Чигарина

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ТЕХНОЛОГИЙ ДОСТУПА К ДАННЫМ В СИСТЕМАХ БАЗ ДАННЫХ

(Самарский национальный исследовательский университет
имени академика С.П. Королёва)

Во всех современных автоматизированных информационных системах используются различные средства для долговременного хранения данных и функции доступа к ним. В настоящее время повышаются требования к скорости обработки данных, и большинство операций нецелесообразно выполнять вручную. Для того, чтобы наиболее полно использовать возможностям того или иного сервера баз данных, необходимо работать с ним напрямую. Это означает полную зависимость приложения от используемого сервера и сложности перехода на другую платформу, так как возникнет необходимость переписывать большое количество кода. Данный вопрос призваны решить различные технологии доступа к данным, являющиеся прослойкой между конкретным сервером и приложением пользователя, предоставляя программисту простой унифицированный механизм работы с данными.

Технологией доступа к данным называется система интерфейсов, обеспечивающая взаимодействие между приложением и базой данных. Во многих системах управления базами данных имеются библиотеки, содержащие интерфейсы прикладного программирования (application programming interface — API), представляющие собой функции, при помощи которых можно выполнять с данными те или иные действия.

На сегодняшний день существует множество технологий доступа к данным, таких как BDE, OLE, ODBC, ADO, и до сих пор разрабатываются новые, более надежные, удобные в работе и более быстродействующие технологии.

На выбор технологии доступа влияет выбор средства разработки приложения и средства реализации базы данных.

На данный момент одними из наиболее популярных и используемых языков программирования являются объектно-ориентированные языки Java и C#. Основным преимуществом языка Java является кроссплатформенность. Язык C# был разработан как язык разработки приложений платформы Microsoft .NET Framework. Платформа .NET основана на использовании общей среды выполнения кода CLR (Common Language Environment), подобно виртуальной Java-