

Идентификация процессов и систем с использованием параллельного имитационного моделирования и нейросетевой аппроксимации

П. В. Назаров, А.М. Поплетеев, В.М. Лутковский

*Белорусский государственный университет
пр. Скорины 4, Минск, Беларусь*

1. Введение

Задача идентификации, является одной из основополагающих во всех областях науки. В последнее время в литературе появилось большое число работ, посвященных методам анализа и идентификации различных процессов и систем с помощью имитационного моделирования, в частности моделирования методом Монте-Карло [1, 2]. Это обусловлено тем, что для построения имитационной модели любой сложной системы достаточно обладать информацией об элементарных процессах, происходящих в ней, и иметь представление об общей структуре системы. В то же время, стандартное математическое моделирование предполагает наличие полного аналитического описания поведения системы и знание законов распределения всех стохастических параметров, используемых при моделировании.

Однако, при попытке определения неизвестных параметров систем с помощью такого подхода, возникает ряд трудностей, наиболее значимыми из которых являются значительные вычислительные затраты при практической реализации имитационного моделирования. Особенно остро это проявляется при попытке использования имитационной модели в так называемом "фитинге", когда изменяя неизвестные параметры пытаются подобрать их таким образом, чтобы поведение модели соответствовало поведению исследуемой системы. В качестве алгоритмов изменения параметров обычно выбирают стандартные методы многопараметрической оптимизации, в которых имитационная модель выступает в качестве стохастической функции, аппроксимирующей результаты эксперимента [1]. Очевидно, что в этом случае число пусков имитационного моделирования совпадает или даже превышает в несколько раз число вычислений функции невязок.

В этой работе предложено несколько путей решения рассмотренной проблемы, в том числе:

- применение параллельных вычислений при имитационном моделировании;
- построение нейросетевой аппроксимации имитационной модели.

Рассмотрим вначале построение нейросетевой аппроксимации, поскольку применение этого подхода отнюдь не исключает, а скорее расширяет возможности для распараллеливания вычислений.

2. Нейросетевая аппроксимация

Рассмотрим следующую задачу. Пусть у некоторой экспериментальной системы имеется набор входных параметров P , часть из которых известна (P_0), а часть предстоит определить (P_X), и набор выходных значений F . В этом случае можно сказать, что рассматриваемая система выполняет некоторое преобразование Θ :

$$\Theta(P_0, P_X) = F. \quad (1)$$

Пусть для этой системы можно построить адекватную имитационную модель. Тогда эта модель будет выполнять преобразование Ξ :

$$\Xi(P_0, P_x) = F. \quad (2)$$

Алгоритм определения параметров с использованием имитационного моделирования выглядит следующим образом:

1. Экспериментально получают некоторую выборку выходных значений F при различных входных параметрах системы.
2. Делаются начальные приближения неизвестных параметров P_x .
3. Запускается алгоритм оптимизации, который, изменяя значения P_x , минимизирует ошибку.

Как уже упоминалось, наиболее серьёзной проблемой в этой схеме являются временные затраты на выполнение имитационного моделирования. В некоторых случаях это приводит к тому, что подобный подход не применим вообще из-за слишком большого (несколько месяцев) времени выполнения алгоритма оптимизации. Для того, чтобы существенно ускорить этот процесс нами было предложено заменить модель искусственной нейронной сетью.

Из литературы известно, что непрерывные функции могут быть с любой, наперёд заданной точностью аппроксимированы линейной комбинацией и суперпозицией сигмоидальных функций, то есть многослойной нейронной сетью [3]. В нашем случае это означает, что операция имитационного моделирования Ξ может быть аппроксимирована нейросетевым преобразованием:

$$\Psi(P_0, P_x) = F^*. \quad (3)$$

Вычислительные затраты при этом будут значительно ниже, чем при имитационном моделировании. Так, в наших экспериментах по моделированию фотофизических процессов, применение нейросетевого моделирования позволило ускорить получение результата в 10^4 – 10^5 раз. Немаловажным фактором в пользу применения нейросетевой аппроксимации является то, что результатом работы нейронной сети является гладкая функция. Это позволяет использовать градиентные методы оптимизации.

Перед началом поиска неизвестных параметров необходимо, используя имитационную модель, создать обучающую выборку и обучить нейронную сеть. Причём, и генерацию обучающей выборки, и обучение приходится повторять при любом изменении модели, например, при её коррекции либо усложнении. Поскольку на генерацию обучающей выборки приходится более 99% всех временных затрат этого метода, было предложено применить инструмент параллельных вычислений, рассмотренный в п.3.

3. Применение параллельных вычислений

Одной из отличительных особенностей имитационного моделирования методом Монте-Карло является высокая степень параллельности вычислений. Так, для получения статистически достоверного результата необходимо произвести от нескольких десятков до сотен тысяч независимых, связанных между собой только общей логикой моделирования, пусков имитационной модели. Такой вычислительный процесс требует больших вычислительных затрат, но поддаётся ускорению путем распараллеливания [4].

Резюмируя всё сказанное выше, заметим, что для ускорения процесса поиска неизвестных параметров можно применить следующие методы (рис. 1):

1. Распараллеливание имитационного моделирования на этапе получения статистически оправданного результата. Этот метод является базовым и наиболее универсальным. Параллельное имитационное моделирование

может быть применено как при идентификации системы, так и на начальном этапе – анализе адекватности модели.

2. Построение нейросетевой аппроксимации позволяет значительно ускорить подбор параметров модели и применять при оптимизации методы градиентного поиска. Параллельные вычисления при этом могут применяться на этапе генерации обучающей выборки.
3. Ещё одним способом, позволяющим ускорить идентификацию, является применение параллельных методов оптимизации, например, генетических алгоритмов. В настоящей работе этот подход не рассматривается.

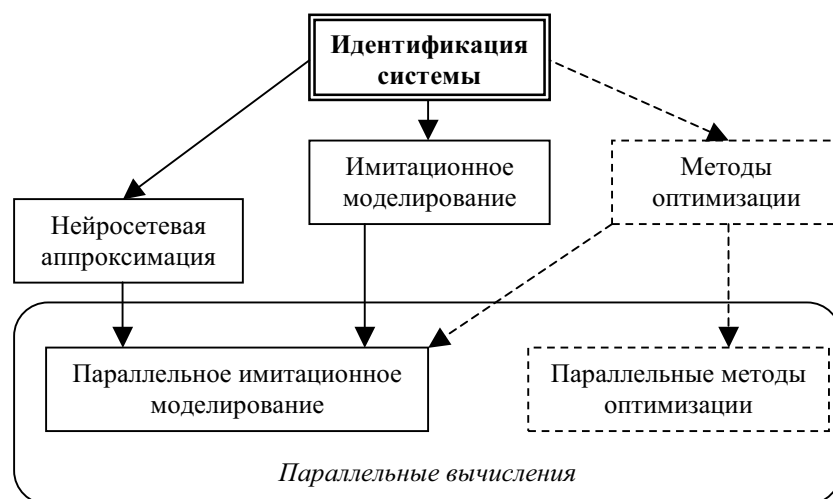


Рис. 1. Параллельные вычисления при идентификации параметров системы с помощью имитационного моделирования.

В результате исследований была создана система параллельного имитационного моделирования, описанная ниже. Этот программный продукт может применяться как на этапе проверки адекватности модели, так и при идентификации параметров некоторой системы.

4. Система параллельного имитационного моделирования ParSim

Программа создана для организации параллельного имитационного моделирования на компьютерах под операционными системами Windows®9X/NT/XP, соединенных в локальную сеть.

Имитационная модель представляется файлом динамически подключаемой библиотеки (dynamic-link library). Помимо самой модели, этот файл содержит также различную служебную информацию: версию файла, описание модели и её параметров, пределы допустимых значений параметров модели и их количество.

Роли в системе распределены следующим образом: на одном из компьютеров запускают серверную часть программы (далее просто "сервер"), с которой будет работать пользователь, на остальных – клиентскую часть (клиенты), которая и выполняет вычисления.

Ниже приведено описание работы программы. После запуска сервера пользователь выбирает файл, содержащий нужную модель. Если загрузка файла модели прошла успешно, в окне программы отображается его версия, а также описание загруженной имитационной модели.

Далее, на остальных компьютерах, предназначенных для проведения распределенных вычислений, и соединенных с сервером локальной сетью, запускается клиентская часть программы. При запуске все клиенты устанавливают соединение с сервером. Чтобы обмен данными по сети не мешал работе программы (как сервера, так и клиента), он реализуется в отдельном потоке. Сервер создает такой поток для каждого клиентского соединения. Это позволяет сделать обмен данными с каждым клиентом независимым от остальных. В течение работы системы сервер поддерживает список всех клиентов. В этом списке содержатся: имя машины, на которой запущен клиент, ее производительность, а также информация о текущем состоянии клиента (принимает параметры, производит вычисления, передает результаты, если произошла ошибка – ее описание, и т.п.). Таким образом, пользователь имеет всю необходимую информацию о каждом клиенте и может контролировать работу системы.

Как только соединение с клиентом было установлено, сервером выполняется инициализация клиента: ему сопоставляется уникальный идентификационный номер, запрашивается имя машины, на которой клиент запущен, указывается полное сетевое имя файла модели. После загрузки модели клиент определяет производительность компьютера, на котором он запущен и отправляет полученный результат на сервер. На этом этап инициализации клиента завершается, и клиент переходит в состояние ожидания сообщений от сервера.

Теперь для каждого клиента пользователь может задать вручную либо загрузить из файла параметры модели, которые сервер затем рассылает клиентам. Также каждому клиенту указывается количество пусков моделирования, так как обсчет каждого набора параметров должен быть осуществлен многократно. Количество пусков имитационного моделирования прямо пропорционально производительности клиентской машины. После того, как все клиенты приняли параметры модели, пользователь может дать команду к началу моделирования, и сервер посылает соответствующее сообщение всем клиентам. Получив от сервера сообщение о начале моделирования, каждый клиент производит обсчет модели согласно полученным параметрам. Поскольку процесс моделирования занимает значительное время, каждый клиент периодически (раз в несколько секунд) отправляет на сервер "I am alive"–сообщение. Это позволяет серверу контролировать активность клиентов. Если клиент в течение долгого времени не передает никаких сообщений, это свидетельствует либо о перегруженности сети, либо о неполадках на клиентской машине.

Если в процессе вычислений происходит критическая ошибка в модельной функции, сообщение, содержащее описание этой ошибки, передается на сервер, и клиент пытается повторить вычисления. Если ошибка повторяется, пользователь может либо изменить параметры модели для этого клиента и снова запустить вычисления, либо разорвать соединение с клиентом.

Если по каким-либо причинам пользователь пожелал прервать процесс моделирования, сервер уведомляет об этом всех клиентов, вычисления прекращаются, и клиенты переходят в режим ожидания команд с сервера.

После выполнения каждого одиночного моделирования клиент отправляет полученные результаты на сервер и, если моделирование было произведено нужное количество раз, переходит в режим ожидания команд с сервера. Такому

клиенту можно передать новый набор параметров и снова запустить процесс моделирования.

Когда все клиенты завершили вычисления, пользователь может сохранить полученные результаты в файл или скопировать их в буфер обмена (clipboard), после чего процесс моделирования может быть повторен для другой модели, либо с иными параметрами.

Все изменения состояния системы, а также процесс передачи данных подробно протоколируется. Пользователь может просмотреть протокол работы, а также сохранить его в файл.

5. Заключение

Разработанные в рамках работы алгоритмы были апробированы на задаче анализа фотофизических процессов в сложных биомолекулярных системах [2]. Применение методов нейросетевой аппроксимации вместе с системой распараллеливания имитационного моделирования позволило ускорить процесс определения параметров таких систем более чем в 10^4 раза.

В дальнейшем планируется применить также параллельные методы оптимизации, а именно, генетические алгоритмы. Это позволит повысить устойчивость процесса поиска неизвестных параметров системы.

Авторы признательны руководству Белорусского государственного университета за финансовую поддержку данного проекта (грант БГУ 540/18).

Литература

1. Yatskou, M.M., Donker, H., Novikov, E.G., Koehorst, R.B.M., van Hoek, A., Apanasovich, V.V., Schaafsma, T.J. "Non-isotropic Excitation Energy Transport in Organized Molecular Systems: Monte Carlo Simulation-based Analysis of Time-resolved Fluorescence" Journal of Physical Chemistry A, Vol. 105(41) (2001) 9498-9508.
2. Nazarov, P.V., Apanasovich, V.V., Lutkovski, V.M., Hemminga, M.A., Koehorst R.B.M. "Neural Network Simulation of Energy Transfer Processes in a Membrane Protein System" in proceedings of the Sixth International Conference on Neural Networks and Soft Computing (ICNNSC), Zakopane, June 11-15, (2002), p. 80.
3. Cybenko G. "Approximations by superpositions of a sigmoidal function" // Math Contr Signals Syst, 1989, Vol. 2, pp. 304-314.
4. Шпаковский. Г.И. Организация параллельных ЭВМ и суперскалярных процессоров: Учеб.пособие. Мн.: Белгосуниверситет, 1996. 284 с.