

---

**АНАЛИЗ РЕЗУЛЬТАТОВ ИМИТАЦИИ И ОПТИМИЗАЦИЯ ИМИТАЦИОННОЙ МОДЕЛИ С ПРИМЕНЕНИЕМ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ****Г. А. Колеватов (Пермь)**

В настоящее время популярность имитационного моделирования растет, увеличивается количество инструментов моделирования и их возможности. По мере роста возможностей инструментов растет и сложность создаваемых моделей. Уже существуют модели, насчитывающие десятки тысяч взаимосвязанных элементов. При использовании таких моделей значительно возрастает сложность имитационного исследования, что приводит к снижению эффективности.

Проблема может быть решена за счет применения методов Data Mining. Конечная цель имитационного исследования заключается в получении знаний о реальной системе, модель которой анализируется [3]. Для достижения этой цели необходимо проанализировать большое количество данных, полученных в результате имитационных экспериментов. Методы Data Mining позволяют выявить взаимосвязи между объектами модели, сократить объем анализируемых данных и снизить трудоемкость анализа.

В качестве отдельной задачи имитационного моделирования часто выделяют построение оптимальной модели. Рядом исследователей [2, 4] предпринимаются попытки по автоматизации процесса оптимизации. Способность методов Data Mining к поиску закономерностей модели могла бы позволить вычислять наиболее оптимальное состояние системы, сохраняя при этом выводы, на основе которых была выполнена оптимизация.

В данной работе авторы излагают свой подход к решению указанных проблем с помощью методов Data Mining на базе системы имитационного моделирования вычислительных сетей Triad.Net. Идея подхода основана на предположении, что зависимость между переменными модели (численными значениями, характеризующими состояния объектов модели) будет выражена в динамике изменения их значений. Получить информацию о зависимости, следовательно, можно с помощью методов Data mining. Полученные данные можно использовать для построения логических выводов о взаимосвязях переменных и для оптимизации системы.

**Анализ результатов в Triad.Net**

Triad.Net – это среда дискретно-событийного моделирования, включающая в себя среду проектирования и создания моделей, компилятор и ядро выполнения модели. Модель в системе автоматизированного проектирования и моделирования Triad можно определить как  $M = (STR, ROUT, MES)$ , где STR – слой структур, ROUT – слой рутин, MES – слой сообщений. Слой структур представляет собой совокупность объектов, взаимодействующих друг с другом посредством посылки сообщений. Слой рутин описывает алгоритмы, по которым объекты взаимодействуют друг с другом. Слой сообщений (MES) предназначен для описания сообщений сложной структуры [1].

Для сбора, обработки и анализа имитационных моделей в системе Triad.Net существуют специальные объекты – информационные процедуры. Информационные процедуры ведут наблюдение за элементами модели, отслеживая состояние наблюдаемого объекта, и сохраняют информацию о его изменении, формируя временной ряд. Следует отметить, что информационные процедуры объявляются независимо от модели. Таким образом, возможно изменение информационных процедур без изменения модели и наоборот.

### Использование средств Data Mining в имитационном моделировании

В 2001 г. был разработан алгоритм автоматической оптимизации модели, основанный на оптимизации целевых функций с помощью эвристических правил [2]. Целевая функция – это некий показатель, содержательно характеризующая модель, значение которой требуется максимизировать/минимизировать. В качестве решения был предложен переборный алгоритм, ограниченный эвристическими правилами.

В 2005 г. был предложен новый способ формирования выходных данных имитационного моделирования [3]. Вместо обычного для моделирующих систем отчета, содержащего статистические данные, применяются методы Data mining для анализа полученных данных. В качестве отчета в этом случае выступают коэффициенты корреляции между переменными. Таким образом, общий объем данных, которые приходилось анализировать пользователю, существенно уменьшается, а информативность отчета повышается.

В 2009 г. в [4] был разработан алгоритм, позволяющий свести деятельность аналитика к выбору одного из трех предложенных программой системой вариантов. Таким образом, требования к квалификации аналитика и трудовые затраты на анализ модели снижаются. Результат достигается преимущественно за счет значительно сужения области применимости – алгоритм рассматривается только в рамках систем массового обслуживания.

Итак, основной проблемой анализа результатов имитационного моделирования в описанных выше работах авторы считают трудоемкость самого процесса анализа и высокие требования к квалификации аналитика. Считается, что эти два фактора существенно снижают популярность моделирования, и делают его фактически недоступным для массового пользователя. В предложенных работах авторы добились значительных успехов для решения проблем, в то же время ряд из них остались нерешенными.

В настоящей работе приводится попытка развития идей этих авторов. Состояние каждого объекта системы может быть представлено численно в виде некоторой переменной модели. Динамика изменения значения этой переменной формирует временной ряд. Если определить зависимости, в которых находятся переменные модели можно строить выводы о зависимостях между объектами, необходимости изменений для оптимизации модели, о применимости модели и многие другие. Оценить зависимости можно с помощью анализа схожести временных рядов (time series similarity analyses).

#### Анализ схожести временных рядов

Временные ряды отображают динамику изменения некоторой величины во времени. В случае, если величины связаны какого рода зависимостями, динамики их изменения должны обнаруживать своего рода сходства. Поиск такого рода сходства называется similarity problem [8].

В 1998 в [5] был предложен свой метод решения similarity problem, основанный на приведении одного ряда к другому с помощью: масштабирования (scale), сдвига (shift) и скользящего среднего (moving average). Количество примененных операций определяло схожесть между рядами. Порядок применения операций выполнялся с помощью эвристических правил [5]. В таком подходе существует недостаток, заключающийся в том, что эвристические правила не имеют рациональных объяснений, и, как следствие, ненадежны и могут содержать ошибки.

Автор решил дополнить предложенный алгоритм. Предлагается выбрать последовательность преобразований с помощью генетических алгоритмов. Генетические алгоритмы в последнее время приобретают все большую популярность благодаря их способности приближенно решать неформальные задачи за приемлемое время [6].

Поскольку порядок операций не важен, хромосому можно описать в виде тройки (Sh, Sc, MA), где Sh – количество операций сдвига, Sc – количество операций масштабирования, MA – количество операций скользящего среднего. Оптимальное решение должно соответствовать двум условиям:

- операции должны приводить один ряд к другому;
- количество операций должно быть минимально возможным.

Используя эти условия можно построить фитнес функцию

$$F = \sqrt{\sum_{j=1}^L (x_j - y_j)^2 + 1 * \sqrt{Sh^2 + Sc^2 + MA^2}},$$

где  $x_j$  – точки преобразуемого ряда,  $y_j$  – точки ряда, к которому выполняется приведение;  $L$  – длина ряда. Такая фитнес-функция позволяет одновременно учесть оба требования. В качестве кроссовера было выбрано универсальное скрещивание и однобитовая мутация.

При проверке алгоритм хорошо показал себя при оценке схожести рядов образованных линейными и тригонометрическими функциями, а также их суперпозицией, содержащей гауссовы шумы. Производительность алгоритма можно оценить как  $O(\log(n)n)$ .

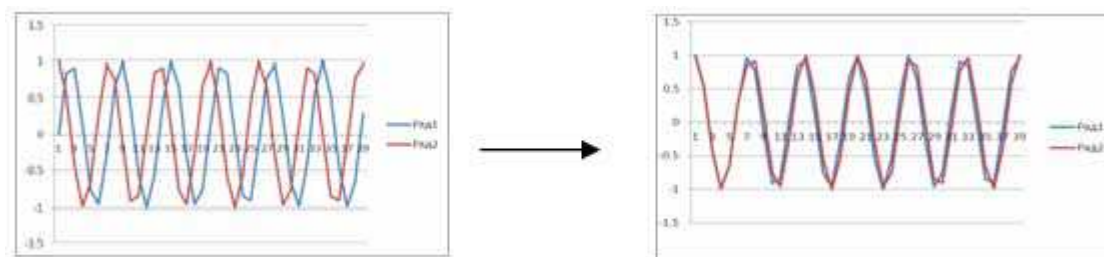


Рис.1. Результат приведения некоторых временных рядов

Для анализа состояния системы мы можем выполнить попарный анализ схожести временных рядов, представляющих информацию о динамике изменения важных величин модели. Результатом такого анализа будет верхне-диагональная матрица, содержащая информацию о схожести значимых величин модели. Пример матрицы представлен в табл. 1.

Таблица 1

#### Результаты имитационного эксперимента

	A	B	C	D
A		8.8891	2.0476	5.1272
B			3.6027	10.0281
C				32.8177
D				

Матрицу схожести также можно представить в виде полно связанного графа, вес ребер которого соответствует показателю схожести временных рядов соответствующих величин. Если мы хотим выяснить величину схожести между величинами A и B, то нам следует учесть, что величина A может влиять на величину B опосредованно.

Таким образом, наиболее оптимальный способ оценить влияние величины A на величину B – найти минимальный путь из A в B. Стоимость этого пути и будет ото-

бражать уточненный *показатель влияния* величин А и В. Пример графа схожести представлен на рис. 2.

Графовое представление результатов анализа открывает новые возможности для анализа моделей, состоящих из нескольких десятков тысяч взаимодействующих элементов. Другое важное достоинство алгоритма состоит в том, что его можно применять для автоматической оптимизации модели.

#### Автоматическая оптимизация модели

Согласно теории анализа результатов имитационного моделирования, для описания работоспособности системы следует использовать показатель(ли) эффективности (performance measure(s)) [7]. Значение этого показателя характеризует эффективность системы. Можно выделить переменные модели, которые зависят только от параметров модели и не зависят от других переменных. Назовем их входными переменными модели. Если отследить зависимость показателя эффективности от входных переменных и выделить переменные, которые влияют на него сильнее, можно сделать вывод о том, какие параметры модели необходимо изменить для ее оптимизации. Следует отметить, что идея схожа с подходом, изложенным в [2]. Ее отличие заключается в том, что выбор параметра для изменения основан на знании о его влиянии на показатель эффективности.

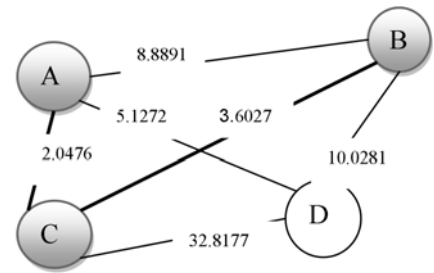


Рис.2. Результат анализа, представленный в виде графа

Таблица 2

#### Матрицы параметров модели

i,j	2А			2Б			2В		
	1	2	3	1	2	3	1	2	3
1	0.33	0.33	0.33	0.9	0.1	0	1	0	0
2	0.33	0.33	0.33	0.1	0.9	0	0	1	0
3	0.33	0.33	0.33	0.5	0	0.5	0.5	0	0.5

Анализ схожести позволяет получить *показатель влияния* переменных модели на показатель эффективности. Переменная, значение показателя влияния для которой минимально, предположительно оказывает наиболее сильное влияние на показатель эффективности. Можно изменять ассоциируемый с входной переменной параметр для достижения более оптимального состояния. После проведения серии изменений можно повторить процесс анализа и выбрать следующий для изменения параметр.

Для наглядности изложения воспользуемся примером задачи по оптимизации моделируемой системы. Пусть имеются 3 сервера ( $s_i, i = \overline{1,3}$ ) получающие от клиентов запросы на обработку. Каждый сервер при получении запроса либо обрабатывает его сам, либо отправляет на обработку другому серверу с некоторой вероятностью. Параметрами модели будут:

- промежутки времени между запросами клиентов:  $t_i^s, i = \overline{1,3}$  (неизменны);
- время обработки запроса  $i$ -м сервером:  $t_i^s, i = \overline{1,3}$  (неизменно);
- вероятности  $P_{ij}$ , которые мы можем изменять.

Таким образом, наша задача состоит в нахождении такой матрицы  $[P_{ij}]$ , которая соответствовала бы оптимальной работе системы. Определим собираемые величины:  $L_i^o$  – длина очереди сервера.  $L_i^c$  – количество необработанных запросов клиента  $C_i$ .  $[L_{ij}^c], i \neq j$  – количество запросов к серверу  $j$  находящихся в очереди ожидания  $i$ -го сервера. Входными

переменными в нашем случае будет  $[L_{ij}^c], i \neq j$ . Показателем эффективности будет соответственно  $L^c = \sum_{i=1}^3 L_i^c$  – суммарное количество необработанных запросов клиентов.

При выбранных значениях  $t_{c_1} = 2, t_{c_2} = 2, t_{c_3} = 2, t_{s_1} = 1, t_{s_2} = 2, t_{s_3} = 4$  и начальных значениях матрицы вероятностей, представленных в табл. 2А, система получила результат, представленный в табл. 2Б.

Оптимизация позволила улучшить значение показателя эффективности с начального значения 4.55 до 1.47, что почти совпадает со значением 1.44, полученным в результате экспертного анализа системы.

### Заключение

1. Представлена попытка оптимизировать процесс имитационного моделирования за счет применения методов Data Mining. Рассмотрена система проектирования и моделирования компьютерных сетей TriadNS. Приводится способ представления имитационной модели и механизм сбора и обработки данных о функционировании имитационной модели.

2. Описывается алгоритм анализа временных рядов, используемый в TriadNS. Алгоритм является расширением идеи [5] с усовершенствованием выбора оптимального пути преобразования с использованием генетических алгоритмов. Указываются ограничения применения алгоритма и возможности дальнейшего анализа результатов моделирования, в том числе и методами теории графов. Приводится алгоритм автоматической оптимизации модели, основанный на идее [2] и дополняющий ее алгоритмом анализа временных рядов, который позволяет выполнять оптимизацию на основе знаний о взаимосвязях между переменными модели.

### Литература

1. **Mikov A. I.** Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems // Gero J.S. and F.Sudweeks F.(eds), Advances in Formal Design Methods for CAD, Preprints of the IFIP WG 5.2 Workshop on Formal Design Methods for Computer-Aided Design, Mexico, Mexico, 1995. P. 105–127.
2. **Brady T., Bowden R.** The effectiveness of generic optimization routines in computer simulation languages. Paper presented at the Industrial Engineering Research Conference 2001, Institute of Industrial Engineers, TX, USA, 2001, {CD\_ROM}.
3. **Brady T., Yellig E.** Simulation Data mining: A new form of simulation output, Paper presented at the 37th Winter Simulation Conference, Orlando, FL, USA, 2005. P. 285–289.
4. **Robinson S.** Automated Analysis of simulation Output Data, Paper presented at the the 37th Winter Simulation Conference, Orlando, FL, USA, 2005. P. 763–770.
5. **Rafiei, D., & Mendelzon.** Efficient retrieval of similar time sequences using DFT, Paper presented at the FODO '98 Conference, Kobe, Japan, 1998. P. 249–257.
6. **Емельянов В. В., Курейчик В. В., Курейчик В. М.** Теория и практика эволюционного моделирования. М.: Физматлит, 2003. 8 с.
7. **Nakayama M. K.** Simulation Output Analysis, Paper presented at the 34th Winter Simulation Conference, Newark, USA, 2002. P. 23–34.
8. **Ye Nong** Handbook of Data Mining, 2003, Lawrence Erlbaum Associates Publishers, London, UK.