

О. В. Кузнецова, И. В. Рудаков

ОПТИМИЗАЦИЯ СТРУКТУРЫ СЛОЖНОЙ ДИСКРЕТНОЙ СИСТЕМЫ, ФОРМАЛИЗОВАННОЙ СЕТЬЮ МАССОВОГО ОБСЛУЖИВАНИЯ

Рассмотрена проблема оптимизации структуры сложной дискретной системы, формализованной сетью массового обслуживания. Предложена модификация генетического алгоритма для решения данной задачи. Приведен пример работы алгоритма, а также оценка эффективности алгоритма.

E-mail: irudakov@yandex.ru

Ключевые слова: сложная дискретная система, сеть массового обслуживания, структурная оптимизация, генетический алгоритм.

В настоящее время актуальной задачей является исследование сложных дискретных систем (СДС), отображающих функционирование таких объектов, как автоматизированные системы управления технологическими процессами, информационно-вычислительные системы, комплексы, сети, транспортные и банковские информационные системы и т.д. В общем виде процесс исследования системы можно представить с помощью IDEF0-диаграммы (рис. 1).

Приведенные на схеме блоки выполняют следующие функции:

- блок “Моделирование поведения системы” служит для анализа работы системы в условиях, максимально приближенных к реальным условиям;
- блок “Сбор статистики” позволяет хранить информацию об особенностях функционирования системы;
- блок “Анализ статистики” предназначен для анализа и изучения информации о системе;
- блок “Оптимизация системы” необходим для модификации системы в соответствии с заданными требованиями к ней;
- блок “Моделирование полученной системы” служит для изучения функционирования модифицированной системы;
- блок “Анализ новой системы” позволяет анализировать информацию о новых свойствах системы, полученных при модификации, а также принимать решение об успешности процесса оптимизации.

Наиболее важными при анализе функционирования СДС являются задачи оптимизации, что объясняется повышением сложности и увеличением размеров исследуемых систем. Можно выделить два типа оптимизации [1] — параметрическая и структурная. С практической точки зрения наибольший интерес представляет задача структурной

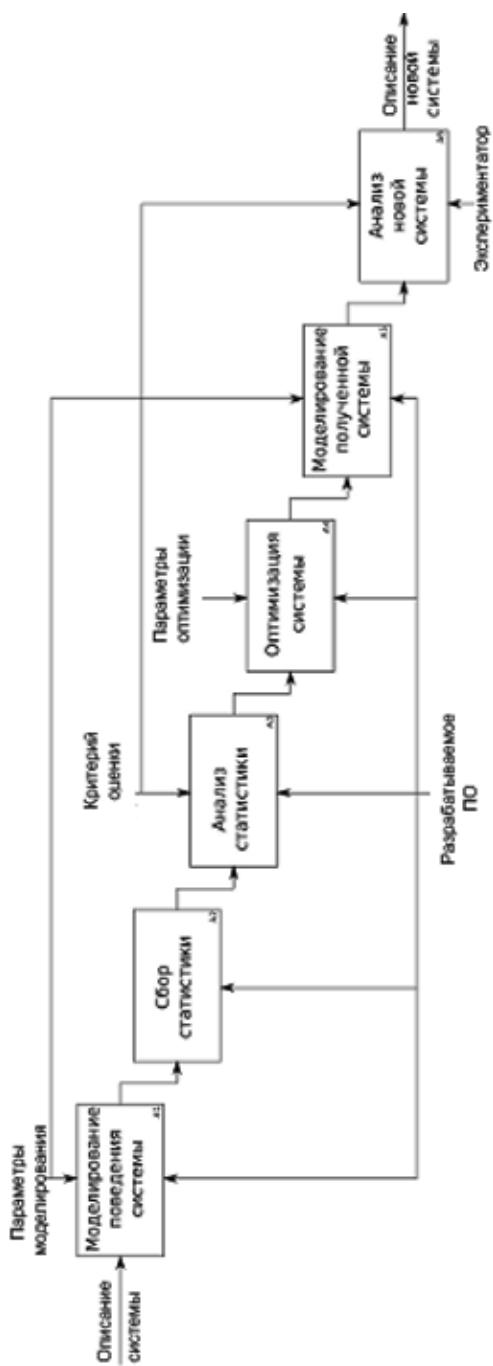


Рис. 1. IDEFO-диаграмма процесса исследования системы

оптимизации [2] как наиболее сложная ввиду того, что она плохо поддается формализации, поскольку свойства исследуемого объекта зачастую зависят от большого числа случайных, противоречивых и мало исследованных факторов.

При решении задачи структурной оптимизации выделяют несколько подходов:

- градиентные численные методы [3];
- методы теории графов [4];
- методы теории принятия решений [5].

Постановка задачи структурной оптимизации

Для постановки задачи оптимизации необходимо определить объект оптимизации и критерии оптимальности [6], т.е. параметры, по которым проводят сравнительную оценку полученных результатов.

Объект оптимизации. Сложные дискретные системы могут быть формализованы в виде различных типовых математических схем, таких как дифференциальные уравнения, конечные или вероятностные автоматы, системы массового обслуживания. В общем виде системы массового обслуживания представляют собой совокупность конечного числа обслуживающих узлов, в которой циркулируют заявки, переходящие в соответствии с маршрутной матрицей из одного узла в другой.

Объектом оптимизации является система массового обслуживания (СМО), которая представляет модель исследуемой структуры. Для обеспечения возможности проведения оптимизации СМО элементы системы должны быть:

- функционально идентичными — выполнять один и тот же набор функций с точки зрения реального объекта;
- взаимозаменяемы — элементы в системе можно менять (добавлять, удалять, переставлять) без нарушения функций системы в целом.

Критерии оптимальности. Среди параметров, характеризующих СМО, выделяют:

среднее время обслуживания одной заявки

$$f_1 = \sum_{i=1}^N t_{\text{busy}_i} + \sum_{i=1}^N t_{\text{wait}_i},$$

где N — число приборов обслуживания, которые будут обрабатывать заявку; t_{busy_i} — время обработки заявки i -м прибором обслуживания; t_{wait_i} — время ожидания обработки i -м прибором обслуживания;

число потерянных заявок

$$f_2 = \sum_{i=1}^N n_{\text{lost}_i},$$

где N — число приборов обслуживания, которые будут обрабатывать заявку; n_{lost_i} — число отказов на i -м приборе обслуживания.

На основании аддитивного способа построения целевой функции [1] ее представляют в виде суммы критериев с нормирующими коэффициентами. Иными словами, целевую функцию для данной задачи записывают как

$$f = a \cdot f_1 + b \cdot f_2, \quad (1)$$

где f_1, f_2 — среднее время обработки заявки и число потерянных заявок соответственно; a, b — нормирующие коэффициенты, задаваемые экспериментатором.

Модифицированный генетический алгоритм. Рассматриваемый в данной статье алгоритм является модификацией генетического алгоритма [7]. Однако следует отметить, что системы, получаемые в процессе эволюции, должны обладать рядом свойств. Во-первых, необходимо, чтобы все исследуемые системы были эквивалентны друг другу по входам и выходам. Во-вторых, существуют количественные ограничения, т. е. минимальная и максимальная размерность системы. Общая схема алгоритма приведена на рис. 2.

Выделяют следующие основные этапы алгоритма.

1. Создание начальной популяции. Данный этап менее всего подвержен автоматизации ввиду того, что создание множества систем,



Рис. 2. Схема алгоритма оптимизации сложной дискретной системы

отвечающих описанным выше требованиям, — процесс трудно формализуемый.

Для дальнейшего применения оператора кроссовера и мутации в целях формирования систем с новой топологией необходимо начальное множество систем (в терминах генетических алгоритмов — особей). В заданном множестве выделяют одну систему — исходную, остальные системы (элементы множества начальной популяции) создаются с использованием экспертной оценки. Размерность начального множества является изменяемым параметром алгоритма.

2. Выбор родительской пары. Поскольку получаемые особи дочерней популяции должны обладать наилучшей приспособляемостью, то они ее наследуют от родительских особей. Существует несколько способов выбора родительской пары [7]. В данном случае для скрещивания выбираются две особи (системы) с максимальным значением приспособляемости (минимальным значением целевой функции).

3. Оператор кроссовера. Данный оператор обеспечивает получение дочерних особей из родительской пары. Алгоритм применения оператора кроссовера состоит из двух шагов.

1-й шаг — определение позиции кроссовера. Во-первых, необходимо отметить, что кодирование хромосом осуществляется посредством матрицы инцидентности. Во-вторых, для облегчения этого кодирования при нумерации элементов системы предпочтение отдается элементам, обеспечивающим реализацию входных потоков заявок, т. е. все генераторы в системе имеют наименьшие порядковые номера. Таким образом, позиция кроссовера определяется как случайное число в интервале

$$[n_{gen} + 1; N - 1],$$

где n_{gen} — количество генераторов в системе; N — размерность системы.

2-й шаг — применение оператора кроссовера. После определения позиции кроссовера выполняется процедура применения данного оператора: перестановка элементов в матрице, т. е. для двух матриц (хромосомы родителей) элементы с индексами, которые по вертикали и по горизонтали больше или равны позиции кроссовера, меняются местами:

$$M_{child1} = [m_{ij}], \text{ где } m_{ij} = \left\{ \begin{array}{l} m_{ij}^{par2}, i \geq P_{cross} \wedge j \geq P_{cross} \\ m_{ij}^{par1}, \text{ иначе} \end{array} \right\};$$

$$M_{child2} = [m_{ij}], \text{ где } m_{ij} = \left\{ \begin{array}{l} m_{ij}^{par1}, i \geq P_{cross} \wedge j \geq P_{cross} \\ m_{ij}^{par2}, \text{ иначе} \end{array} \right\};$$

M_{child1} , M_{child2} — хромосомы дочерних особей; M_{par1} , M_{par2} — хро-



Рис. 3. Применение оператора кроссовера

мосомы родительских особей; m_{ij}^{par1} , m_{ij}^{par2} — элементы хромосом родительских особей; P_{cross} — позиция кроссовера.

Описанная выше схема показана на рис. 3.

4. Наследование. Данный оператор относится к дополнительным операторам генетического алгоритма. После получения дочерних систем в них могут отсутствовать входные потоки у ряда элементов, поэтому для них предлагается наследовать элементы, обеспечивающие недостающие входные потоки. Иными словами, к множеству элементов системы E добавляются такие элементы из множества E_p родительской системы, которые удовлетворяют следующему утверждению: для элементов x_1, x_2 , таких, что $x_1 \in E_p \wedge x_2 \in E \wedge x_2 \in E_p$ существует поток заявок $\lambda: x_1 \rightarrow x_2$. Следовательно, в множество E добавляется элемент x_1 . Этот процесс повторяется итерационно до тех пор, пока множество не станет транзитивно-замкнутым. В результате все элементы множества E будут иметь входящие потоки заявок.

5. Улучшение геномина. Данный оператор является дополнительным. Он также предназначен для устранения из системы элементов с отсутствующими входными потоками. Из множества E элементов системы итерационно удаляются все элементы x , для которых не может быть найдено ни одного элемента $y \in E$ — такого, чтобы существовал поток $\lambda: y \rightarrow x$.

Размерность системы после выполнения данного оператора будет невелика по сравнению с размерностью системы, полученной путем использования оператора наследования. Следует отметить, что применение этих двух дополнительных операторов является взаимоисключающим.

6. Селекция. Существует много способов отбора особей [1] для включения в популяцию. В данном случае дочерние особи включаются в репродукционную группу, из которой отбирается N “лучших” (с наибольшей приспособляемостью, или минимальным значением целевой функции) особей для создания популяции.

7. Критерий завершения. Работа алгоритма завершается, когда разность между целевыми функциями на соседних итерациях становится меньше заданного значения точности:

$$|f - f_{\text{last}}| < \varepsilon,$$

где f — текущее значение целевой функции; f_{last} — значение целевой функции с предыдущей итерации; ε — заданная точность.

Программное обеспечение для реализации алгоритма. Программное обеспечение, реализующее данный алгоритм, состоит из шести блоков:

блок оптимизации — реализуется алгоритм оптимизации, а информация о вновь полученной системе собирается с помощью блока моделирования;

блок моделирования — реализуется алгоритм Дельфт [8], обеспечивающий процесс имитации работы системы, при этом полученная информация сохраняется в блоке сбора статистики;

блок сбора статистики — является хранилищем информации о функционировании систем, анализируемых в процессе оптимизации;

блок “Хранение системы” — является хранилищем информации о системах, элементах системы (законах функционирования, максимальной длине очереди и пр.) и связях между данными элементами;

блок работы с файлами — дополнительный блок, не участвующий непосредственно в процессе оптимизации, но выполняющий вспомогательную функцию загрузки в файл информации о системе, а также о сохранении информации о процессе оптимизации, результирующей системе, ее функционировании;

блок графического интерфейса — необходим для задания системы и получения графической информации о результирующей системе.

Пример работы и результаты экспериментов. Приведем пример работы алгоритма оптимизации СДС, оценки зависимостей целевой функции от размерности системы и числа итераций алгоритма от размера начальной популяции.

На рис. 4 показаны исходная популяция (система A и ее аналоги B и C , заданные экспертом) и полученная в результате работы описанного выше алгоритма система D с улучшенными характеристиками. Система создана за 27 итераций алгоритма при начальных параметрах: точность — 0,1; вероятность наследования — 0,4; максимальный размер популяции — 5. Законы функционирования элементов данных систем описаны в таблице.

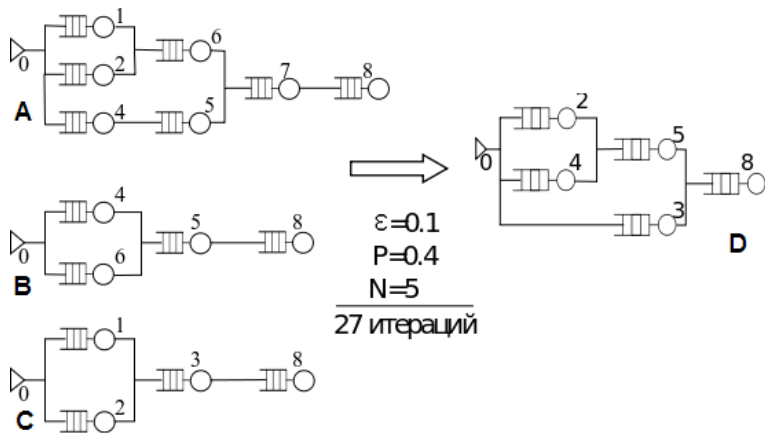


Рис. 4. Пример преобразования системы

Описание элементов системы

Элемент	Закон функционирования
0	R(0,2)
1	Exp(2)
2	Exp(3)
3	R(3,5)
4	R(1,2)
5	N(4,1)
6	N(3,0.5)
7	R(2,5)
8	N(3,0.25)

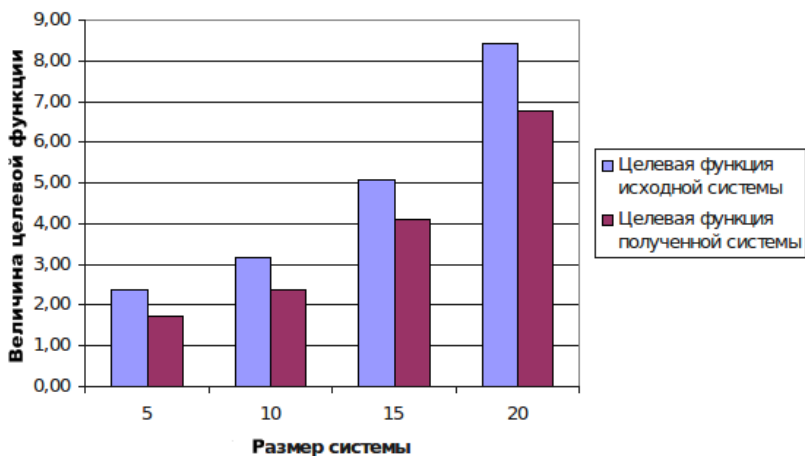


Рис. 5. Зависимость целевой функции от размера системы

При исследовании системы произвольной топологии получена зависимость между целевой функцией и размером системы (рис. 5). Среднее значение улучшения системы (по значению целевой функции) составляет 15–20 %.

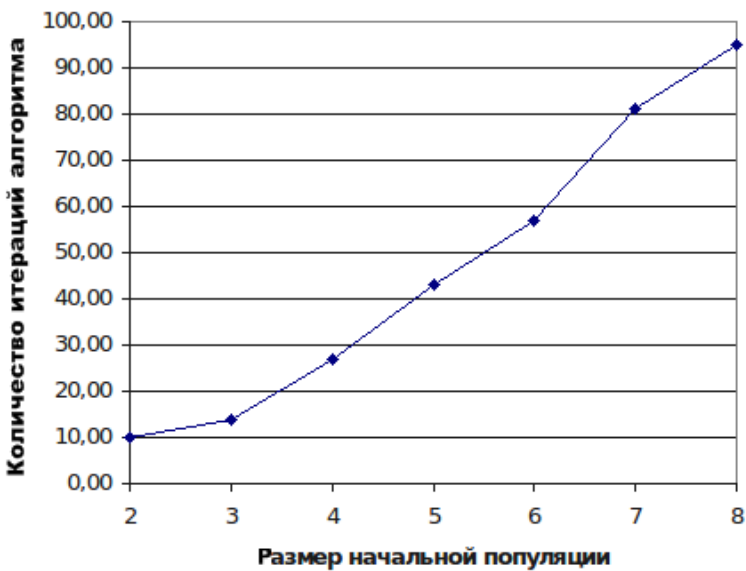


Рис. 6. Зависимость числа итераций алгоритма от размера начальной популяции

Размер начальной популяции влияет на число итераций алгоритма (рис. 6). По результатам исследования данной зависимости, которая характеризует работу алгоритма, можно сделать следующие выводы:

- число итераций прямо пропорционально зависит от размера системы;
- зависимость числа итераций от размера начальной популяции имеет вид, представленный на рис. 6, ввиду того, что большее количество начальных систем обеспечивает большую глубину поиска наилучшего решения.

Таким образом, предложенный алгоритм для оптимизации сложных дискретных систем, формализованных сетями массового обслуживания, позволяет получать системы с наилучшими параметрами функционирования.

СПИСОК ЛИТЕРАТУРЫ

1. Н о р е н к о в И. П. Основы автоматизированного проектирования: учеб. для вузов. – 3-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2006. – 448 с.
2. Б о ж к о А. Н., Т о л п а р о в А. Ч. Структурный синтез на элементах с ограниченной сочетаемостью // Методолог [Электронный ресурс] / ред. А. Кудрявцев. – URL: <http://www.metodolog.ru/00562/00562.html/> – [Режим доступа: свободный].
3. З а д о р о ж н ы й В. Н., Е р ш о в Е. С., К а н е в а О. Н. Двухуровневые градиентные методы для оптимизации сетей с очередями // Омский научный вестник. – 2006. – № 7 (43). – С. 119–126.

4. О в ч и н н и к о в В. А. Алгоритмизация комбинаторно-оптимизационных задач при проектировании ЭВМ и систем: Учеб. для вузов. – М.: Изд-во МГТУ им. Н.Э.Баумана, 2001. – 288 с.
5. Б у г р о в Д. А. Постановка задачи структурной оптимизации магистральной корпоративной телекоммуникационной сети // Информатика и космос. – 2005. – № 2. – С. 42–47.
6. А т т е т к о в А. В., Г а л к и н С. В., З а р у б и н В. С. Методы оптимизации: учеб. для вузов / Под ред. В.С. Зарубина, А.П. Крищенко. – 2-е изд., стереотип. – М.: МГТУ им. Н.Э. Баумана, 2003. – 440 с.
7. Г л а д к о в Л. А., К у р е й ч и к В. В., К у р е й ч и к В. М. Генетические алгоритмы: Учеб. пособие. – М.: Физматлит, 2006. – 320 с.
8. Т е х н о л о г и я системного моделирования / Е.Ф. Аврамчук, А.А. Вавилов, С.В. Емельянов, В.В. Калашников. – М.: Машиностроение; Берлин: Техник, 1988. – 520 с.

Статья поступила в редакцию 10.05.2012