

# Использование параллельных вычислений для решения прикладной задачи оптимизации компьютерной сети с помощью генетических алгоритмов

А.С. Татаренко

*Московский государственный университет экономики,  
статистики и информатики (МЭСИ)  
tatarenko\_84@mail.ru*

Построение оптимальной структуры коммуникационной сети – важное направление исследований, возникновение которого можно связать с возникновением коммуникационных сетей более 100 лет назад. У данного направления определенно есть хорошее будущее, учитывая скорость увеличения количества передаваемого сетевого трафика по всему миру. Так, по прогнозам компании CISCO[1] в 2013 году объемы глобального IP-трафика вырастут в 5 раз, его объем достигнет 56 экзбайт в месяц (в 2008-м было всего 9).

Построение коммуникационной сети включает в себя огромное количество конкретных задач. В статье рассматривается одна из них. На начальном этапе построения сети возникает задача оптимального размещения сетевых узлов, которые будут соединены с конечными пользователями (для удобства именуемых далее по тексту «точки доступа») и между собой. Необходимо соединить (прямо или через промежуточные узлы) все точки доступа. При этом оптимизируется стоимость построения сети при известных стоимостях установки узла, соединения узла и точки доступа и соединения 2 произвольных узлов. Для простоты в рассматриваемой модели наложено условие, что все узлы попарно соединены, а также, что расположение каждого узла совпадает с какой-либо точкой доступа. При этом не рассматривается плотность передаваемых потоков данных, так как на начальном этапе построения сети крайне мало сведений о реальных нуждах пользователей сети.

Математически задача формулируется следующим образом [2]:

$$F = \sum_{i=1}^N \sum_{j=i}^N \xi_{ij} u_{ij} + \sum_{j=1}^N \eta_j r_j \rightarrow \min, \quad (1)$$

где:

$i, j = 1, 2, \dots, N$  - расположение точки доступа\узла;

$\xi_{ij}$  - стоимость соединения узла\точки доступа  $i$  и узла\точки доступа;

$\eta_j$  - стоимость «открытия» узла;

$K_j$  - максимальное количество точек доступа, которые можно соединить с  $j$ -м узлом;

$u_{ij} = 1$ , если узел\точка доступа  $i$  соединена с узлом\точкой доступа  $j$ ; иначе 0;

$r_j = 1$ , если расположен узел; иначе 0.

(С учетом ограничений, описанных выше).

Представленная задача относится к классу сложных вычислительных задач линейного программирования. Она может быть решена методами линейного программирования, разнообразными схемами эвристического поиска или с помощью генетических алгоритмов. В работе рассматривается использование генетических алгоритмов, так как такой подход позволяет эффективно работать с задачами большой размерности, а также обеспечивает гибкий подход к решению, так как слабо зависит от флуктуаций целевой функции.

Оптимизационный процесс на основе генетических алгоритмов представляет собой постепенное улучшение целевой функции по мере появления новых представителей генофонда. Воспроизводство новых особей происходит следующим образом.

1. Из текущей популяции наугад выбирается пара особей, причем вероятность выбора каждой из них пропорциональна значению целевой функции данной особи.
2. С некоторой вероятностью (задана константой в приведенном исследовании) происходит так называемая «рекомбинация» (также известна как «кроссовер») для выбранной пары, либо воспроизводство идентичной пары.
3. С некоторой вероятностью (также заданной константой в данном исследовании) происходит мутация воспроизведенных особей.
4. Новые особи, созданные из текущей популяции, в совокупности представляют собой новую популяцию и процесс повторяется с шага 1 [4].

Рекомбинация или кроссовер представляет собой составление новой особи из «кусков» генетической информации родительских особей. Генетическая информация каждой особи хранится в виде строки битов. Для целей данного исследования применяется односточный кроссовер, суть которого заключается в следующем: для родительских хромосом (т. е. строк) случайным образом выбирается точка раздела, и они обмениваются отсеченными частями. Полученные две строки являются потомками:

$$\begin{array}{l} 11010\ 01100101101 \Rightarrow 10110\ 01100101101 \\ 10110\ 10011101001 \Rightarrow 11010\ 10011101001 \end{array}$$

Мутация особи представляет собой случайное инвертирование каждого бита в генетическом коде особи с некоторой вероятностью (в данном исследовании задано константой).

Задача относится к классу NP-полных и ее сложность стремительно растет с ростом размерности. В этой связи целесообразным представляется использование параллельных вычислений. Использовать параллельные вычисления для данной задачи можно несколькими способами:

1. Каждый отдельный вычислительный кластер решает задачу выбора и скрещивания особей из одной популяции. Таким образом, несколько вычислительных машин будут совместно формировать поколение за поколением.
2. Каждый кластер формирует отдельную популяцию. Между полученными популяциями затем происходят различного рода миграции генов, как то: случайная выборка; пропорциональный отбор, т.е. для миграции выбираются наиболее приспособленные особи (имеющие наилучшую функцию полезности). При этом можно некоторым способом ограничить возможные переходы, т.е. разрешить переходы особей между одними популяциями и запретить между другими, сформировав таким образом некоторую топологию переходов между имеющимися популяциями.
3. Возможно разделение имеющихся вычислительных мощностей по классам задач. Например, применение алгоритма «рабочий-хозяин»[5]. Компьютеры-рабочие реализуют процессы воспроизводства, мутации и вычисления функции пригодности (целевой функции в рассматриваемой задаче), а компьютер-хозяин хранит текущие популяции и проводит отбор в новые популяции.

Предпочтительным для решения поставленной задачи является 2й способ, так как он не требует больших затрат вычислительного ресурса для передачи данных между кластерами, и в то же время обеспечивает лучшее использование имеющейся генетической информации.

Программный код модели написан на языке Java с использованием базы данных Oracle XE. Проведен анализ скорости «улучшения» популяции, т.е. улучшения целевой

функции ее особей, в зависимости от нескольких факторов: численность первоначальной популяции, вероятность кроссовера, вероятность мутации, вероятность выбора особи из текущей популяции в зависимости от значения целевой функции, а также различные способы миграции особей между популяциями.

Таким образом, представленная модель на основе генетического алгоритма позволяет эффективно решать задачу оптимизации топологии компьютерной сети, с использованием параллельных вычислений.

### **Литература**

1. Новостной портал ITUA.info.
2. Pioro M., Medhi D. "Routing, Flow, and Capacity Design in Communication and computer networks", Elsevier, 2004 г. ISBN: 0-12-557189-5.
3. David E. Goldberg "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 2006 г., ISBN 0-20-115767-5
4. Melanie Mitchel "An Introduction to Genetic Algorithms", The MIT Press, Cambridge, Massachusetts, 1998 г., ISBN 0-262-13316-4
5. Панченко, Т.В. «Генетические алгоритмы», учебно-методическое пособие, Издательский дом «Астраханский университет», 2007