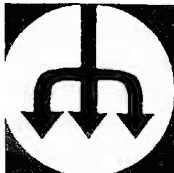


12. Плотников В. В. Изменчивость ледяного покрова Берингова моря // Метеорология и гидрология. 1997. № 7. С. 53—60.
 13. Сергин В. Я., Сергин С. Я. Системный анализ проблемы больших колебаний климата и оледенения Земли. Л.: Гидрометеоиздат, 1978. 279 с.
 14. Соболев И. М., Статников Р. Б. Выбор оптимальных параметров в задачах со многими критериями. М.: Наука, 1981. 110 с.
 15. Стронгин Р. Г. Численные методы в многоэкстремальных задачах (информационно-статистические алгоритмы). М.: Наука, 1978. 239 с.
 16. Тимохов Л. А., Хейсин Д. Е. Динамика морских льдов (математические модели). Л.: Гидрометеоиздат, 1987. 272 с.
 17. Фукунага К. Введение в статистическую теорию распознавания образов. М.: Наука, 1979. 368 с.

18. Четырбоцкий А. Н. Формирование и разрушение ледяного покрова замерзающих морей // Криосфера Земли. 2005. Т. 9. № 3. С. 45—53.
 19. Четырбоцкий А. Н. Прогноз ледового режима на акватории Японского моря (математическая модель) // Информационные технологии. 2005. № 8. С. 9—14.
 20. Четырбоцкий А. Н., Плотников В. В. Ледяной покров Японского моря: анализ данных и моделирование. Владивосток: Дальнаука, 2005. 146 с.
 21. Чупрынин В. И. Разрывные автоколебания в геофизических системах. М.: Наука, 1985. 93 с.
 22. Global Digital Sea Ice Data Bank (GDSIDB). http://dmiweb.dmi.dk/pub/GDSIDB_mirror/
 23. Thorndike A. S. et al. The thickness distribution of sea ice // J. Geophys. Res. 1975. Vol. 80. N 33. P. 4501—4513.



ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

УДК 004.7:004.021

Р. М. Алгулиев, д-р техн. наук,
Р. М. Алыгулиев, канд. физ.-мат. наук,
М. Г. Шарифов,

Институт информационных технологий НАН
 Азербайджана
 rasim@science.az, a.gamiz@science.az,
 master@mprojection.net

Подход к оптимальному расположению серверов CDN в узлах глобальных сетей

В целях минимизации стоимости доставки Web-контента конечным потребителям предлагается модель оптимального расположения серверов CDN в узлах глобальных сетей, которая реализуется в два этапа. На первом этапе на основе степени важности Web-контента осуществляется кластеризация узлов, а на втором этапе предлагается алгоритм определения оптимальных точек расположения серверов CDN. Кластеризация узлов и выбор оптимальных точек расположения серверов CDN сводятся к задачам комбинаторной оптимизации, которые решаются с помощью генетических алгоритмов.

Введение

С быстрым ростом числа клиентов в структуре Web, имеющей архитектуру клиент—сервер, возникает серьезная проблема, состоящая в том, что Web-сервер быстро перегружается, что, в свою очередь,

приводит к ухудшению QoS (*Quality of Service*) — показателя пользователей, и увеличению трафика на Интернет. Практическое решение этой проблемы, реализуемое различными способами, состоит в том, чтобы реплицировать сервер и получить вместо одного централизованного сервера множество серверов. Существуют различные технологии репликации [1], среди которых наиболее широко распространенной является сеть доставки контента (*Content Delivery Network* — CDN) [2—4].

CDN — это множество распределенных серверов, которые, кэшируя или сохраняя Web-контент, обеспечивают их доставку географически распределенным пользователям. В таких сетях Web-контент однократно размещается в серверах CDN, откуда доставляются конечным потребителям. Когда поступает запрос от пользователей, CDN переадресует его к ближайшему серверу, и, следовательно, за счет доставки информации с ближайших серверов уменьшается время ответа, запрашиваемое Интернет-пользователями. Доставка Web-контента пользователям с ближайших серверов позволяет снизить и трафик на Интернет. По сути, CDN является наложенной сетью, которая оперирует на верхних уровнях сети Интернет, следовательно, она также решает еще одну немаловажную проблему — снижение капитальных затрат поставщиков информации, т. е. позволяет отказаться от создания собственной дорогостоящей сетевой инфраструктуры. Благодаря перечисленным преимуществам начался бурный рост индустрии CDN [5, 6] уже с момента появления первого оператора (провайдера) CDN [7].

При проектировании CDN провайдером приходится решать ряд сложных проблем, среди которых доминирующую роль играют выбор оптимальных точек расположения серверов CDN в узлах глобальных сетей [8–11] и оптимальное размещение Web-контентов в серверах CDN [12–14]. При выборе оптимальных точек расположения серверов CDN и оптимальном размещении Web-контентов провайдером CDN следует учитывать интерес каждого узла. Когда число узлов глобальной сети и Web-контентов достаточно велико, учитывать интересы узлов очень трудно. Поэтому в данной работе, прежде чем определить оптимальные точки расположения серверов CDN, осуществляется кластеризация узлов. При этом каждый узел представляется как множество значений Web-контентов, где в качестве значений берутся веса Web-контентов в узлах. После кластеризации на каждом кластере определяются оптимальные точки расположения серверов CDN. Кластеризация узлов и выбор оптимальных точек расположения сводятся к задаче комбинаторной оптимизации. Для решения задачи комбинаторной оптимизации предлагаются генетические алгоритмы. Теперь переходим к описанию предложенной модели.

Математическая формулировка задачи кластеризации узлов

Пусть задана глобальная сеть $N = (U, C)$, где $U = (u_1, \dots, u_N)$ — множество узлов, $C = (c_{jl})$, c_{jl} — стоимость передачи единичной информации между узлами j и l , $c_{jl} = c_{lj} > 0$, $1 \leq j \neq l \leq N$, $c_{jj} = 0$, $j = 1, \dots, N$. Далее, пусть $WC = (wc_1, \dots, wc_M)$ представляет множество Web-контентов, которые часто запрашиваются узлами глобальной сети, и $Ser = (ser_1, \dots, ser_S)$ — множество серверов CDN, которые должны быть расположены в узлах глобальной сети таким образом, чтобы при размещении Web-контентов в этих узлах общая стоимость их доставки конечным потребителям была минимальной. Предполагается, что $M > N > S$.

Введем генное выражение (*gene-based expression*) узлов глобальной сети. Генное выражение — это представление объектов (узлов) как множество значений признаков (Web-контентов), т. е. $u_j = (w_{1j}, \dots, w_{Mj})$, где w_{ij} — вес Web-контента wc_i в узле u_j , $i = 1, \dots, M$, $j = 1, \dots, N$, алгоритм вычисления которого предлагается ниже.

Очевидно, что каждый Web-контент имеет локальный и глобальный веса в узлах глобальной сети. Для определения этих весов используются формулы, предложенные в работе [5], которые были введены для вычисления локального и глобального весов слов в предложении. Введем некоторые предварительные обозначения: пусть n_{ij} — число

обращений к Web-контенту wc_i , инициированных в узле u_j ; N_j — общее число обращений ко всем Web-контентам $WC = (wc_1, \dots, wc_M)$, инициированных в узле u_j ; M_i — общее число обращений к Web-контенту wc_i , инициированных во всех узлах сети, $i = 1, \dots, M$; $j = 1, \dots, N$.

С учетом введенных обозначений локальный вес Web-контента wc_i в узле u_j будет определяться формулой

$$w_{ij}^{\text{local}} = \log_2 \left(1 + \frac{n_{ij}}{N_j} \right), \quad (1)$$

а глобальный вес по формуле

$$w_{ij}^{\text{global}} = 1 - \varepsilon_i. \quad (2)$$

В последней формуле через ε_i обозначена нормализованная энтропия, определяемая следующей формулой:

$$\varepsilon_i = - \frac{1}{\log_2(N)} \sum_{j=1}^N \frac{n_{ij}}{M_i} \log_2 \left(\frac{n_{ij}}{M_i} \right). \quad (3)$$

Вес w_{ij} Web-контента wc_i в узле u_j равняется произведению локального (1) и глобального (2) весов:

$$w_{ij} = w_{ij}^{\text{local}} w_{ij}^{\text{global}}, \quad (4)$$

где $i = 1, \dots, M$; $j = 1, \dots, N$.

После такого представления нам остается применить методы *data mining* с целью обнаружения естественных групп среди узлов сети. Одной из основных задач *data mining* для обнаружения естественных групп является кластеризация [16]. Цель задачи кластеризации заключается в разбиении узлов заданной сети на непересекающиеся q кластеры $Q = (Q_1, Q_2, \dots, Q_q)$ таким образом, чтобы узлы в одном кластере были более близки друг к другу, чем узлы в разных кластерах.

Таким образом, каждый узел идентифицирован характеристическим вектором Web-контентов, элементы которых определяются формулой (4). При таком представлении легко можно вычислить меру подобия, т. е. вес связи между узлами глобальной сети. Мера подобия между узлами u_j и u_k определяется метрикой косинуса:

$$\text{sim}(u_j, u_k) = \cos(u_j, u_k) = \frac{\sum_{i=1}^M w_{ij} w_{ik}}{\sqrt{\sum_{i=1}^M w_{ij}^2} \sqrt{\sum_{i=1}^M w_{ik}^2}}, \quad j, k = 1, \dots, N, \quad j \neq k. \quad (5)$$

Из формулы (5) видно, что большое значение $\text{sim}(u_j, u_k)$ будет соответствовать большей степени подобия узлов u_j и u_k . Тогда оптимальную кластеризацию узлов, т. е. группировку подобных узлов, можно достичь путем максимизации суммы

$$\sum_{u_j, u_k \in C_p} \text{sim}(u_j, u_k) \rightarrow \max \quad (6)$$

в каждом кластере $Q_p, p = 1, \dots, q$.

Введем булево переменное x_{jp} , равное единице, если узел u_j относится к кластеру Q_p , и равное нулю в противном случае:

$$x_{jp} = \begin{cases} 1, & \text{если } u_j \in Q_p, \\ 0, & \text{если } u_j \notin Q_p. \end{cases} \quad (7)$$

Тогда задачу (6) можно сформулировать так:

$$f_1(x) = \sum_{p=1}^q \sum_{j=1}^{N-1} \sum_{k=j+1}^N s_{jk} x_{jp} x_{kp} \rightarrow \max. \quad (8)$$

В формуле (8) для краткости записи принято обозначение $s_{jk} = \text{sim}(u_j, u_k)$.

При кластеризации предполагается, что кластеры не пересекаются $Q_p \cap Q_r = \emptyset (p \neq r)$, и поэтому переменные x_{jp} должны удовлетворять ограничениям вида

$$\sum_{p=1}^q x_{jp} = 1, \quad j = 1, \dots, N. \quad (9)$$

Таким образом, задача кластеризации узлов состоит в нахождении максимума суммы (8) при ограничениях (7) и (9).

Выбор оптимальных точек расположения CDN серверов

После кластеризации следующим шагом является определение оптимальных CDN узлов на каждом кластере. При этом основная цель заключается в размещении каждого из M Web-контентов в одном из серверов CDN таким образом, чтобы суммарные стоимости доставки Web-контентов между узлами были минимизированы. Каждое такое размещение представляет собой перестановку (m_1, m_2, \dots, m_M) , составленную из чисел $(1, 2, \dots, N_p)$, $m_i \in \{1, 2, \dots, N_p\}, i = 1, \dots, M$, где N_p — число узлов в кластере Q_p . При этом любое размещение Web-контента wc_i в узел m_i описывается соответствием $i \rightarrow m_i$. Например, при $M = 10$ и $N_p = 3$ перестановка $(1, 1, 2, 3, 2, 2, 3, 1, 1, 3)$ соответствует тому, что

Web-контенты $wc_i (i = 1, 2, 8, 9)$ размещены в первом, $wc_i (i = 3, 5, 6)$ — во втором, а $wc_i (i = 4, 7, 10)$ — в третьем узлах. При любом размещении Web-контента wc_i в узел m_i и Web-контента wc_k в узел m_k мы имеем стоимость передачи информации между узлами m_i и m_k , которая равна произведению потока v_{m_i, m_k} (за v_{m_i, m_k} берется среднее значение потока информации между узлами m_i и m_k за определенный промежуток времени) на стоимость c_{m_i, m_k} передачи единичной информации между ними. Таким образом, задача сводится к нахождению перестановки (m_1, m_2, \dots, m_M) чисел $(1, 2, \dots, N_p)$, минимизирующей суммарные стоимости:

$$\sum_{i=1}^{M-1} \sum_{k=i+1}^M v_{m_i, m_k} c_{m_i, m_k}. \quad (10)$$

Пусть y_{ij_p} — бинарное переменное, равное единице, если Web-контент wc_i размещен в узле j_p , равное нулю — в противном случае. С учетом последнего обозначения формула (10) записывается в виде

$$f_2(y) = \sum_{i=1}^{M-1} \sum_{j_p=1}^{N_p} \sum_{k=i+1}^M \sum_{l_p=1}^{N_p} c_{j_p, l_p} v_{j_p, l_p} \times \\ \times y_{ij_p} y_{kl_p} \rightarrow \min. \quad (11)$$

Если предположить, что каждый Web-контент назначается только одному из узлов CDN, то должно выполняться условие

$$\sum_{j_p=1}^{N_p} y_{ij_p} = 1, \quad i = 1, \dots, M. \quad (12)$$

Ограниченность объема памяти серверов требует выполнение следующего условия:

$$\sum_{i=1}^M y_{ij_p} v_i \leq z_{j_p} V_{j_p}, \quad j_p = 1, \dots, N_p, \quad (13)$$

где v_i — объем Web-контента wc_i ; V_{j_p} — объем памяти сервера, размещенного в j_p -м узле.

Переменное z_{j_p} определяется таким образом:

$$z_{j_p} = \begin{cases} 1, & \text{если узел } j_p \text{ выбран как GDN узел,} \\ 0 & \text{— в противном случае.} \end{cases}$$

Следует учесть, что серверы CDN располагаются не в каждом узле:

$$\sum_{j_p=1}^{N_p} z_{j_p} = S_p, \quad (14)$$

где S_p означает число серверов CDN, предназначенных для размещения в узлах кластера Q_p . При предположении $S > q$, число S_p можно определить формулой

$$S_p = \text{int} \left(\frac{N_p}{N} \right) S,$$

где $\text{int}(a)$ означает целую часть, меньшую либо равную числу a .

Согласно определению переменных y_{ij_p} и z_{j_p} :

$$y_{ij_p} \in \{0, 1\} \quad \forall i, j_p; \quad (15)$$

$$z_{j_p} \in \{0, 1\} \quad \forall j_p; \quad (16)$$

$$y_{ij_p} \leq z_{j_p} \quad \forall i, j_p. \quad (17)$$

Задача (11)—(17) позволяет выбрать оптимальные точки расположения серверов CDN на каждом кластере $Q_p, p = 1, \dots, q$, при размещении Web-контентов в которых обеспечивается минимизация суммарной стоимости доставки Web-контентов конечным потребителям.

Для определения оптимальных точек размещения серверов CDN в других кластерах предлагается следующая стратегия. Web-контенты, размещенные в серверах CDN кластера Q_p , удаляются из дальнейшего рассмотрения, и задача (11)—(17) в других

кластерах решается с меньшим числом $M - \sum_{p=1}^{q-1} M_p$

Web-контентов, где M_p — число Web-контентов, раз-

мещенных в CDN серверах кластера $Q_p, M = \sum_{p=1}^q M_p$.

Такой подход позволяет на каждом шаге уменьшить размерность задачи (11)—(17), которая на p -м

шаге будет равняться числу $\left(M - \sum_{k=1}^{p-1} M_k \right) N_p$. Лег-

ко доказать, что общее число переменных по всем шагам q будет меньше, чем число переменных задачи (11)—(17) без осуществления кластеризации,

$\sum_{p=1}^q \left(M - \sum_{k=1}^{p-1} M_k \right) N_p < MN$. Это является непо-

средственным подтверждением эффективности предложенного подхода.

Очевидно, что (8), (7), (9) и (11)—(17) относятся к задачам комбинаторной оптимизации. Известно, что при больших значениях N, q и M (в реальном случае так и есть) решение таких задач требует больших вычислительных затрат. Для преодоления этой трудности предлагается использовать генетические алгоритмы, которые являются эффективными инструментами при решении задач дискретной оптимизации большой размерности. Генетические алгоритмы — это алгоритмически оптимизированные стратегии, базирующиеся на принципах естественной эволюции. Множество потенциальных решений проблемы конкурирует друг с другом, при этом улучшается общее качество решения подобно тому, как этот процесс проходит в популяции живых организмов [17].

Замечание 1. При размещении Web-контентов в серверах CDN предполагается выполнение усло-

вия $\sum_{i=1}^M v_i \leq \sum_{s=1}^S V_s$. В противном случае, определив

степень важности Web-контента wc_i в узле u_j

$\text{score}_{ij} = \frac{n_{ij}}{N_j} : \frac{M_i}{M_\Sigma}$ ($i = 1, \dots, M; j = 1, \dots, N$), из мно-

жества $WC = (wc_1, \dots, wc_M)$ по заданному порогу τ выбираются те Web-контенты, которые удовлетворяют следующим условиям [18]:

$$\exists wc_i \text{ score}_{ij} \geq \tau \text{ и } n_{ij} \geq 2;$$

$$\sum_{j=1}^N 1\{n_{ij} > 0\} \geq 2,$$

где через M_Σ было обозначено общее число обращений, инициированных во всех узлах глобальной сети.

Генетический алгоритм решения задачи (8), (7), (9)

Инициализация и предварительная обработка начальной популяции. Генетический алгоритм, состоящий из нескольких шагов, начинается с кодирования решений в виде хромосом. Хромосомы, представляющие решения задачи, конструируются в таком виде: $X_{K_x} = [A_{K_x, 1}, A_{K_x, 2}, \dots, A_{K_x, N}]$, где $K_x = 1, \dots, P_x, N$ — длина хромосомы, а P_x — размер популяции. Здесь A_{K_x, J_x} и J_x ($J_x = 1, \dots, N$), соответственно, означают значение и локус (позицию) гена. A_{K_x, J_x} принимает значения в интервале $[1, q]$. При таком кодировании узлам будут соответствовать локусы, а кластерам — значения генов. Согласно кодированию, решение задачи (8), (7), (9) декодирует-

ся по формуле $x_{j, A_{K_x, J_x}} = \begin{cases} 1, & \text{при } j = J_x \\ 0, & \text{при } j \neq J_x \end{cases}, j = 1, \dots, N.$

Например, пусть $q = 3, N = 8$, тогда кодирование $X = [2, 3, 1, 3, 1, 1, 2, 3]$ соответствует тому решению, когда переменные $x_{31}, x_{51}, x_{61}, x_{12}, x_{72}, x_{23}, x_{43}, x_{83}$ равны единице, а переменные $x_{K_x, 1}$ ($K_x = 1, 2, 4, 7, 8$), $x_{K_x, 2}$ ($K_x = 2, 3, 4, 5, 6, 8$), $x_{K_x, 3}$ ($K_x = 1, 3, 5, 6, 7$) равны нулю, т. е. $u_3, u_5, u_6 \in Q_1, u_1, u_7 \in Q_2$ и $u_2, u_4, u_8 \in Q_3$.

Применение генетического алгоритма к задачам оптимизации с ограничениями сталкивается с проблемой появления недопустимых решений. Недопустимыми являются решения, нарушающие ограничения задачи, которые, в основном, появляются при генерации начальной популяции. Такое кодирование при генерации начальной популяции позволяет избегать проверки допустимости хромосом и, следовательно, ускоряет процесс сходимости генетического алгоритма. Следует отметить, что на скорость сходимости и оптимальность найденного решения влияет и другой фактор — разновидность хромосом в начальной популяции. Разновидность в популяции можно получить путем удаления идентичных хромосом. При этом основная трудность заключается в определении метрики, определяющей меру близости хромосом. Мера близости хромосом определяется расстоянием Хемминга. Расстояние Хемминга между двумя хромосомами X_{K_x} и X_{L_x} вычисляется по формуле

$$H(X_{K_x}, X_{L_x}) = \sum_{J_x=1}^N E_{J_x},$$

где

$$E_{J_x} = \begin{cases} 0, & \text{если } A_{K_x, J_x} = A_{L_x, J_x}, \\ 1, & \text{если } A_{K_x, J_x} \neq A_{L_x, J_x}. \end{cases} \quad (18)$$

Легко видеть, что расстояние Хемминга принимает максимальное, равное N , значение, когда значения генов на соответствующих позициях разные, и наоборот, принимает минимальное, равное 0, значение, когда значения генов на соответствующих позициях равные. Следовательно, $H(X_{K_x}, X_{L_x}) \in [0, N]$.

С помощью метрики Хемминга новая популяция из начальной популяции получается таким способом. Вычисляется расстояние Хемминга между всевозможными парами хромосом в начальной популяции, и составляется список хромосом, в которых расстояние между ними и другими хромо-

сомами близко к минимальному расстоянию в популяции. В этом списке хромосомы с худшим *fitness* значением удаляются из популяции.

Разновидность популяции оценивается величиной

$$H^{avg} = \frac{1}{P_x(P_x - 1)} \sum_{K_x=1}^{P_x-1} \sum_{J_x=K_x+1}^{P_x} H(X_{K_x}, X_{L_x}),$$

где большое значение H^{avg} означает, что хромосомы в популяции резко отличаются друг от друга и, наоборот, малое значение соответствует тому, что хромосомы в популяции близки друг другу.

Оператор отбора. Следующим шагом генетического алгоритма является отбор хромосом для скрещивания. Прежде чем выбрать родительские хромосомы они упорядочиваются в порядке убывания (или возрастания) относительно их *fitness* значений. Пусть $F_1 = (f_{1,1}, f_{1,2}, \dots, f_{1,P_x})$ — упорядоченный список значений функции приспособленности (здесь в качестве функции приспособленности взята целевая функция $f_1(x)$). Ясно, что при таком упорядочении хромосома, стоящая на первом (последнем) месте, является наилучшей. Другими словами, при возрастании (убывании) рангов степень пригодности хромосом уменьшается. Для выбора родительских хромосом предлагается такое правило: из интервала $[1, P_x]$ с равномерной вероятностью выбираются два числа $P_{x,1}, P_{x,2}$, потом на основе этих чисел из упорядоченного списка выбирается та хромосома, ранг которой равен числу $P_x^s = \min(P_{x,1}, P_{x,2})$ (или $P_x^s = \max(P_{x,1}, P_{x,2})$), т. е. из двух хромосом выбирается наилучшая. Следовательно, при таком выборе хромосома с лучшим *fitness* значением имеет наибольший шанс быть отобранной.

Оператор скрещивания. При использовании оператора скрещивания могут появиться хромосомы-дети идентичных хромосом-родителей. Чтобы предотвратить появление таких хромосом, для каждой пары хромосомы-родителя вычисляется бинарный вектор, где j -й элемент определяется формулой (18), и точка скрещивания выбирается между первой и последней единицей. Например, для хромосом $X_1 = [1, 2, 4, 5, 6, 3, 2, 1]$ и $X_2 = [1, 2, 3, 1, 6, 1, 2, 1]$ бинарный вектор будет иметь вид $[0, 0, 1, 1, 0, 1, 0, 0]$, и, следовательно, точками скрещивания могут быть выбраны позиции 4 и 5.

Оператор мутации. Для выхода из локального экстремума оператор мутации является очень эффективным, при котором случайно выбирается хромосома, и изменяется значение случайно выбранного гена. Применение оператора мутации

может привести к тривиальному случаю, когда все узлы будут отнесены одному кластеру. Во избежание появления таких решений, для каждой хромосомы, отобранной для мутации, составляется список индексов кластеров, которые не присутствуют в этой хромосоме, и значение случайно выбранного гена заменяется индексом, взятым из этого списка. После мутации вычисляется новое *fitness* значение и сравнивается со старым *fitness* значением, если новое *fitness* значение является лучшим, чем старое, то в популяции старая хромосома заменяется новой, если наоборот — то в популяции сохраняется старая хромосома.

Критерий завершения. Последним шагом является критерий завершения генетического алгоритма, ко-

торый определяется параметром $\sigma_1 = \frac{f_1^{\max} - f_1^{avg}}{f_1^{\max} - f_1^{\min}}$, яв-

ляющимся вариацией *fitness* значений, где f_1^{\max} ,

f_1^{\min} и f_1^{avg} — максимальное, минимальное и среднее *fitness* значения. Очевидно, что когда хромосомы сосредотачиваются вокруг хромосомы, имеющей лучшее (максимальное) *fitness* значение, параметр приближается к нулю. И следовательно, критерием завершения генетического алгоритма может служить минимальное значение параметра σ_1 .

Генетический алгоритм решения задачи (11)–(17)

Инициализация и предварительная обработка начальной популяции. Как и в предыдущем разделе, решение кодируется в таком виде: $Y_{K_y} = [B_{K_y, 1}, B_{K_y, 2}, \dots, B_{K_y, M}]$, где $B_{K_y, J_y} \in [1, N_p]$, $K_y = 1, \dots, P_y$, $J_y = 1, \dots, M$, M — длина хромосомы, P_y — размер популяции. Аналогично декодированию переменных x_{j_p} декодируются и переменные

$y_{ij_p}, y_{i, B_{K_y, J_y}} = \begin{cases} 1, & \text{при } i = J_y; \\ 0, & \text{при } i \neq J_y, \end{cases} i = 1, \dots, M$. Чтобы

удовлетворить условию (14), инициализацию начальной популяции следует провести в такой последовательности: сначала из интервала $[1, N_p]$ с равномерной вероятностью выбираются числа (в количестве S_p), потом среди них генерируются значения генов B_{K_y, J_y} . Такое кодирование и инициализация обеспечивают выполнение ограничений (12), (15) и (16). При генерации начальной популяции основная трудность заключается в обеспечении выполнения условия (13). Выполнение условия (13) можно обеспечить разными способа-

ми. Одним из широко распространенных способов является проверка допустимости хромосом после каждой генерации. В этом случае генерация продолжается до тех пор, пока не получена допустимая хромосома. При этом может случиться так, что время, потраченное на проверку допустимости хромосом, будет сравнимо со временем решения задачи. Поэтому целесообразно использовать метод штрафной функции. Идея метода штрафной функции заключается в выявлении недопустимых хромосом в популяции. Штрафная функция строится следующим образом:

$$\pi(y) = \sum_{j_p=1}^{N_p} \max\{0, \theta(y \cdot j_p)\},$$

$$\text{где } \theta(y \cdot j_p) = \sum_{i=1}^M y_{ij_p} v_i - z_{j_p} V_{j_p}.$$

Легко можно видеть, что при выполнении ограничения (13) функция $\pi(y)$ принимает нулевое значение, а в противном случае — положительное значение. Итак, начальная популяция очищается от хромосом с положительным значением штрафной функции.

Оператор отбора. Обеспечение разновидности популяции и отбор хромосом осуществляются методами, изложенными в предыдущем разделе.

Оператор скрещивания. Скрещивание хромосом проводится по такой схеме: для каждой пары хромосом-родителя составляются два вектора. Элементы первого вектора состоят из тех аллелей первой хромосомы, которые не присутствуют в другой хромосоме. Аналогично первому строится и второй вектор. Элементы каждого вектора являются потенциальными генами для скрещивания. Количество обменивающихся генов определяется числом r , которое случайно выбирается из интервала $[1, d - 1]$, где d — число элементов в векторах. Например, для хромосом-родителей $Y_1 = [3, 1, 7, 8, 2]$, $Y_2 = [4, 5, 2, 10, 7]$ соответствующие векторы будут $G_1 = [1, 3, 8]$, $G_2 = [4, 5, 10]$ и, следовательно, $d = 3$. Пусть $r = 2$, это означает, что из каждого вектора для обмена выбираются два случайных числа. Пусть этими числами на первом векторе будут 1 и 3, а на втором векторе — 4 и 5, тогда после обмена могут появиться хромосомы-дети: $Y_1^c = [5, 4, 7, 8, 2]$ и

$Y_2^c = [1, 3, 2, 10, 7]$. Такой алгоритм предотвращает появление хромосом-детей, идентичных хромосомам-родителям.

Оператор мутации. Мутация хромосом производится по алгоритму, изложенному в предыдущем разделе.

Критерий завершения. Критерий завершения генетического алгоритма определяется параметром

$$\sigma_2 = \frac{F_2^{\max} - F_2^{\text{avg}}}{F_2^{\max} - F_2^{\min}},$$
 где функция приспособленности

$F_2(y)$ равняется обратному значению целевой

$$\text{функции } F_2(y) = \frac{1}{f_2(y)}.$$

Заключение

Эффективное использование сетевых ресурсов и задача улучшения QoS-показателя пользователей привели к появлению технологии CDN. CDN нацелена ускорить доставку Web-контентов, уменьшить загрузку оригинальных серверов и трафик на Интернет. CDN является сложной системой, при проектировании которой требуется решать множество задач, среди них наиболее актуальными и сложными являются выбор оптимальных точек расположения серверов CDN и оптимальное размещение Web-контентов. Последние годы эти задачи привлекают пристальное внимание многих исследователей и в той или иной мере решаются. Как показали результаты исследования, задачи расположения серверов CDN и размещения Web-контентов решаются в следующих частных случаях: для сетей с древовидной архитектурой; в целях заметного сокращения расстояния до клиентов и простой политики маршрутизации и т. д. При расположении серверов CDN в узлах глобальных сетей провайдеры сталкиваются с проблемой, связанной с размерностью, т. е. провайдерам приходится удовлетворить интерес огромного числа клиентов. Исходя из сказанного, в работе излагается математическая модель оптимального расположения серверов CDN в узлах глобальных сетей, реализация этой модели обеспечивает минимизацию стоимости доставки Web-контентов конечным потребителям. Предложенная модель основывается на кластеризации узлов и выборе оптимальных точек расположения серверов CDN на каждом кластере. Показано, что кластеризация узлов является эффективной, так как позволяет уменьшить число операций. Кластеризация узлов и выбор оптимальных точек расположения серверов CDN сводятся к задачам дискретной оптимизации, для решения которых предлагаются эффективные генетические алгоритмы. Как известно, недопустимые хромосомы появляются не только при генерации начальной популяции, они могут появляться и в результате операторов скрещивания и мутации, несмотря на допустимость хромосом-родителей. Исходя из этого, в работе предложены операторы скрещивания и мутации, которые не нарушают структуру

хромосом. Иными словами, их применение к недопустимым хромосомам не приводит к появлению подобных хромосом. Поэтому для предотвращения появления недопустимых хромосом в шагах скрещивания и мутации предлагается схема для очистки начальной популяции. В противном случае при скрещивании и мутации каждый раз пришлось бы проверять допустимость хромосом, а это потребовало бы немало вычислительных затрат. Следовательно, предложенная стратегия — очистка начальной популяции и операторы скрещивания и мутации — повышает эффективность генетического алгоритма.

Список литературы

1. Таненбаум Э., ван Стен М. Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
2. Vakali A., Pallis G. Content Delivery Networks: Status and Trends // IEEE Internet Computing. 2003. Vol. 7. N 6. P. 68–74.
3. Gadde S., Chase J., Rabinovich M. Web Caching and Content Distribution: A View from Interior // Computer Communications. 2001. Vol. 24. N 2. P. 222–231.
4. Sivasubramanian S., Szymaniak M., Pierre G., van Steen M. Replication for Web hosting systems // ACM Computing Surveys (CSUR). 2004. Vol. 36. N 3. P. 291–334.
5. <http://www.digitalisland.com>
6. Hosanagar K., Krishnan R., Smith M., Chuang J. Optimal pricing of content delivery network (CDN) services // Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS'04). Island of Hawaii. January 5–8, 2004. Track 7. P. 70205a.
7. <http://www.akamai.com>
8. Bassali H. S., Kamath K. M., Hosamani R. B., Gao L. Hierarchy-aware algorithms for CDN proxy placement in the Internet // Computer Communications. 2003. Vol. 26. N 3. P. 251–263.
9. Starobinski P., Wu T. Performance of server selection algorithms for content replication networks // NETWORKING 2005: 4th International IFIP-TC6 Networking Conference. Waterloo. Canada. May 2–6, 2005. Lecture Notes in Computer Science. Springer-Verlag. 2005. Vol. 3462. P. 443–454.
10. Pan J., Hou Y. T., Li B. An overview of DNS-based server selections in content distribution // Computer Networks. 2003. Vol. 43. N 6. P. 695–711.
11. Bakiras S., Loukopoulos T. Combining replica placement and caching techniques in content distribution networks // Computer Communications. 2005. Vol. 28. N 9. P. 1062–1073.
12. Laoutaris N., Telelis O., Zissimopoulos V., Stavrakakis I. Local utility aware content replication // NETWORKING 2005: 4th International IFIP-TC6 Networking Conference. Waterloo. Canada. May 2–6, 2005. Lecture Notes in Computer Science. Springer-Verlag. 2005. Vol. 3462. P. 455–468.
13. Tse S. S. H. Approximation algorithms for document placement in distributed Web servers // IEEE Transactions on Parallel and Distributed Systems. 2005. Vol. 16. N 6. P. 489–496.
14. Loukopoulos T., Ahmad I., Papadias D. An overview of data replication on the Internet // Proceedings of the 6th International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN'02). Manila. Philippines. May 22–24, 2002. P. 31–36.
15. Yeh J.-Y., Ke H.-R., Yang W.-P., Meng I.-H. Text summarization using a trainable summarizer and latent semantic analysis // Information Processing and Management. 2005. Vol. 41. N 1. P. 75–95.
16. Grabmeier J., Rudolph A. Techniques of cluster algorithms in data mining // Data Mining and Knowledge Discovery. 2002. Vol. 6. N 4. P. 303–360.
17. Michalewicz Z. Genetic algorithms + data structures = evolution programs. Berlin: Springer-Verlag, 1996. 387 p.
18. Niu Z.-Y., Ji D.-H., Tan C.-L. Document clustering based on cluster validation // Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM'04). Washington. USA. November 8–13, 2004. P. 501–506.