

С.Д. Погорелый, Р.В. Билоус

Генетический алгоритм балансировки нагрузки в сети

Предложен генетический алгоритм решения задачи балансировки нагрузки на каналы связи в компьютерных и транспортных сетях как многопараметрической задачи оптимизации. Обоснованы особенности представления хромосомом и генетических операций кроссовера, мутации и отбора. Сформированы подходы использования описанного алгоритма в существующих сетях с использованием современных протоколов маршрутизации.

A genetic algorithm is suggested of solving the problem of a balancing of the load on the communication channels in computer and transport networks as a multi-parametric problem of optimization. The peculiarities of the representation of chromosomes and genetic operations of a crossover. Mutation and a selection are substantiated. The approaches of using the described algorithm in the existing networks are formulated with the use of contemporary protocols of the routing.

Запропоновано генетичний алгоритм розв'язання задачі балансування навантаження на канали зв'язку в комп'ютерних та транспортних мережах як багатопараметричної задачі оптимізації. Обґрунтовано особливості представлення хромосом та генетичних операцій кросоверу, мутації та відбору. Сформовано підходи використання описаного алгоритму в існуючих мережах із використанням сучасних протоколів маршрутизації.

Введение. Задача оптимизации транспорта данных (маршрутизация и балансировка нагрузки на каналы связи) стала существенной составляющей современных компьютерных сетей и распределенных вычислительных систем. Ограничения существующих алгоритмов маршрутизации не позволяют достаточно эффективно решить эту задачу в динамических гетерогенных транспортных системах. В современных сетях появилась необходимость решения задачи оптимизации передачи данных (трафика) по многим критериям. Динамика сетей требует учета изменения нагрузки на отдельные каналы связи и формирование механизмов управления процессом маршрутизации в сети в целом. Новые подходы к распределенным вычислениям (облачные вычисления) и создание сложных кластерных высокопроизводительных архитектур (*HPC*) выдвигают дополнительные требования к механизмам маршрутизации и балансировки нагрузки в таких системах. Классические алгоритмы не применимы к задачам такого типа, принадлежащим к классу *NP*-сложных [1]. Возникает необходимость формирования новых подходов и алгоритмов балансировки нагрузки на каналы связи в сети. Один

из таких подходов – создание генетического алгоритма [2].

Постановка задачи

В задаче балансировки нагрузки сеть представлена в виде взвешенного ориентированного графа $G = (V, E, C)$, где V – множество вершин, $E \in V \times V$ – множество ребер графа, $C_{ij}: E \rightarrow R$ – пропускная способность каждого соединения (ребра графа). Информация о потоке между каждой парой вершин задается функцией F_{ij} , где i – исходная вершина, j – вершина назначения. Предложенный алгоритм может быть применен и для отдельной части сети. В таком случае используется определенный подграф графа G .

Генетический алгоритм балансировки нагрузки использует таблицу маршрутизации сети, содержащую m_{ij} альтернативных путей между каждой парой вершин (i, j) . Эта таблица хранит все промежуточные вершины пути $(i \rightarrow j)$, а также «цену» каждого маршрута исходя из алгоритма маршрутизации. Такой подход позволяет применять генетический алгоритм в современных сетях, используя маршруты между двумя вершинами с различной метрикой и административной дистанцией [3].

Во время функционирования сети каждый маршрут между вершинами i и j используется с определенной вероятностью $\alpha_k(i, j)$, причем

* **Ключевые слова:** маршрутизация, генетический алгоритм, задача балансировки нагрузки в сети.

$$\sum_k \alpha_k(i, j) = 1, k = 1 \dots m_{i,j}. \quad (1)$$

Степень использования пропускной способности определенного соединения можно вычислить так:

$$P(l, m) = \frac{1}{C_{l,m}} \sum_k \sum_{i,j \in V'} X_{l,m}^k(i, j) \alpha_k(i, j) F_{i,j},$$

где

$$X_{l,m}^k(i, j) = \begin{cases} 1, & \text{если } (l, m) \in (i \rightarrow j)_k \\ 0, & \text{если } (l, m) \notin (i \rightarrow j)_k \end{cases}. \quad (2)$$

Пусть $P_{\max} = \max_{(l,m) \in E} (P(l, m))$, тогда задача балансировки нагрузки сводится к минимизации

наибольшего потребления канала передачи данных в сети, т.е. к поиску такой конфигурации маршрутов передачи данных, для которой P_{\max} будет наименьшим:

$$\Phi = \min(P_{\max}). \quad (3)$$

Генетический алгоритм

Хромосома определенного поколения генетического алгоритма Y состоит из последовательности аллелей $A_{i,j}$, которые в свою очередь содержат наборы коэффициентов $\alpha_k(i, j)$ для каждой пары вершин $(i, j) \in V'$, т.е.

$$A_{i,j} = \{\alpha_k(i, j)\}. \quad (4)$$

Множество $V' \subset V$ состоит из всех пар вершин, для которых применяется алгоритм балансировки. Для повышения эффективности работы генетического алгоритма и уменьшения затрат на вычисление целесообразно также ограничить множество связей сети (ребер графа G), для которых применяется генетический алгоритм. Обозначим это множество $E' \subset E$. Такое упрощение позволяет выделить критические элементы сети и значительно улучшить динамические характеристики алгоритма. Предложенное представление обеспечивает одинаковое количество и положение аллелей в хромосомах произвольного поколения (рис. 1).

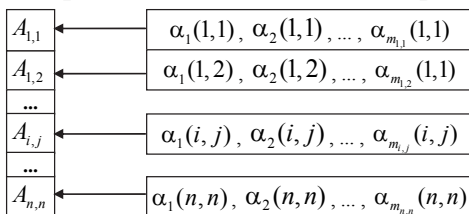


Рис. 1. Представление хромосомы

Начальная популяция

Формирование начальной популяции хромосом зависит от выбранного алгоритма маршрутизации. Пусть для каждой пары вершин $(i, j) \in V'$ существует $m_{i,j}$ альтернативных маршрутов $(i \rightarrow j)$. Пользуясь метрикой алгоритма маршрутизации, каждому маршруту $(i \rightarrow j)_k$ ставится в соответствие определенная величина $Q_{i,j}^k$, характеризующая цену этого пути с учетом алгоритма маршрутизации [4]. При применении нескольких алгоритмов маршрутизации эта величина зависит от метрики маршрута и административной дистанции протокола маршрутизации. Коэффициенты $\alpha_k(i, j)$ начальной популяции генетического алгоритма выбираются так:

$$\alpha_k(i, j) = \frac{Q_{i,j}^k}{\sum_k Q_{i,j}^k}. \quad (5)$$

Следует заметить, что, применяя алгоритм, уместно использовать определенные дополнительные условия для соотношения коэффициентов $\alpha_k(i, j)$:

$$\Omega(\alpha_k(i, j)). \quad (6)$$

Предложенные дополнительные условия накладывают ограничения на отношение величин потоков данных для различных маршрутов в сети. Однако такие ограничения необходимы для уменьшения возможного негативного влияния алгоритма балансировки на динамику передачи данных между определенными точками в сети.

Мутация и кроссовер

Операция мутации для предложенного алгоритма заключается в случайном изменении определенного аллеля хромосомы $A_{i,j}$ (одноточечная мутация) или изменении нескольких аллелей хромосомы одновременно (многоточечная мутация). Операция мутации аллелей в хромосоме должна удовлетворять условиям (1) и (6). Для выбора точек мутации используется эвристический подход. Поскольку задача алгоритма состоит в минимизации наибольшего потребления каналов в сети, лишь изменение величины нагрузки на канал с наибольшим по-

треблением может улучшить решение задачи в следующем поколении генетического алгоритма. Пусть в пределах текущего поколения наибольшее потребление канала наблюдается для канала $(l, m) \in E'$. Следовательно мутации подлежат только те аллели $A_{i,j}$, для которых: $\exists k \in (1 \dots m_{ij})$, такое, что

$$\alpha_k(i, j) \neq 0 \text{ и } X_{l,m}^k(i, j) = 1. \quad (7)$$

Аналогично выбираются и точки кроссовера. В силу предложенного способа представления решения длина хромосомы останется постоянной, поэтому в описанном алгоритме операция кроссовера заключается в обмене частями или отдельными аллелями родительских хромосом [5].

При формировании операций кроссовера и мутации следует учитывать два аспекта:

- сохранение последовательностей аллелей, соответствующих положительному приросту функции приспособленности – для обеспечения сходимости алгоритма в целом;

- эффективное создание новых последовательностей аллелей для обеспечения разнообразия хромосом в пределах поколения и избежания сходимости к локальному экстремуму целевой функции.

Обеспечение описанных требований осуществляется путем адаптивного изменения вероятностей операций кроссовера и мутации хромосом генетического алгоритма [6].

Определим вероятности кроссовера и мутации произвольной хромосомы Y :

$$p_c = \begin{cases} p_1 (F_{\max} - F(Y)) / (F_{\max} - F(Y)), & F(Y) \geq \bar{F} \\ p_2, & F(Y) < \bar{F} \end{cases}. \quad (8)$$

$$p_m = \begin{cases} p_3 (F_{\max} - F(Y)) / (F_{\max} - F(Y)), & F(Y) \geq \bar{F} \\ p_4, & F(Y) < \bar{F} \end{cases}. \quad (9)$$

F_{\max} – значение функции приспособленности лучшей хромосомы популяции, \bar{F} – среднее значение функции приспособленности хромосом в популяции $0 < p_1, p_2, p_3, p_4 \leq 1$.

Для обеспечения разнообразия хромосом в пределах популяции вероятность кроссовера и мутации наименее приспособленных хромосом должна быть высокой, поэтому в предложенном алгоритме $p_1 = p_2 = 1$, $p_3 = p_4 = 0,5$. Такой

адаптивный подход к операциям кроссовера и мутации обеспечивает эволюцию популяции и улучшает сходимость алгоритма. С другой стороны, при росте количества одинаковых хромосом в популяции вероятности мутации и кроссовера растут. Таким образом, предложенный алгоритм позволяет избежать преждевременной сходимости к локальному экстремуму.

Отбор

Поскольку равенство (3) соответствует задаче поиска минимума, функция приспособленности хромосомы Y может быть вычислена как:

$$F(Y) = \left(\max_{(l,m) \in E} (P(l, m)) \right)^{-1}. \quad (10)$$

Алгоритм использует пропорциональный метод отбора в сочетании с методом сохранения лучших особей популяции (элитизмом). Значит, вероятность отбора определенной хромосомы пропорциональна значению функции приспособленности этой хромосомы. Пусть размер популяции N , а значение функции приспособленности хромосомы Y_i , равно $F(Y_i)$, тогда вероятность отбора Y_i вычисляется так:

$$p_i^s = F(Y_i) \cdot \left(\sum_i F(Y_i) \right)^{-1}. \quad (11)$$

Поскольку при описанном подходе существует вероятность статистической ошибки и потери лучших хромосом, в предложенном алгоритме применяется метод сохранения лучших решений. При этом гарантируется переход лучших особей следующего поколения генетического алгоритма и сходимость алгоритма в целом.

Маршрутизация

После завершения работы генетического алгоритма балансировки нагрузки его результаты должны быть использованы в процессе маршрутизации сети. В современных протоколах процесс маршрутизации потока данных между вершинами i и j на промежуточных вершинах (маршрутизаторах) базируется на информации о вершине назначения j пакета данных, который проходит через промежуточную вершину. Поскольку алгоритм балансировки базируется на информации о всех промежуточных верши-

нах маршрутов, следует расширить классический формат таблицы маршрутизации и алгоритма маршрутизации в целом.

Рассмотрим таблицу маршрутизации на промежуточной вершине k сети. В предлагаемом подходе каждая запись таблицы, соответствующая маршруту $(i \rightarrow j)$, который проходит через вершину k , дополнительно содержит информацию о начальной вершине i . Если существует несколько альтернативных путей между вершинами i и j , то запись также содержит вероятность выбора определенного маршрута. Пусть существует m путей $(i \rightarrow j)$, проходящих через вершину k , тогда вероятность выбора некоторого маршрута определяется так:

$$\beta_s(i, j) = \frac{\alpha_s(i, j)}{\sum_s \alpha_s(i, j)}, \quad s = 1 \dots m. \quad (12)$$

Таким образом, результаты работы генетического алгоритма балансировки нагрузки в сети интегрируются в дополнительные записи таблицы маршрутизации в каждой промежуточной вершине обмена трафиком. Алгоритм маршрутизации пакета данных в данном случае выглядит так:

- поиск набора маршрутов, соответствующих информации о вершине назначения пакета данных;
- отбор маршрутов, соответствующих информации о начальной вершине;
- применение определенного маршрута на основе вероятности его выбора.

Существующие записи таблицы маршрутизации не меняются, обеспечивая работу сети в

отсутствие алгоритма балансировки и в случае, когда промежуточная вершина или определенный маршрут не используются алгоритмом балансировки. В таком случае описанный алгоритм выбора маршрута соответствует классическому алгоритму.

Заключение. Описанный подход применим для построения и оптимизации транспортных и компьютерных сетей. Его внедрение позволит более эффективно использовать ресурсы сетей и значительно улучшить динамические характеристики систем балансировки нагрузки в распределенных вычислительных системах.

1. Girish M., Zhou B., Hu J. Formulation of the load engineering problems in MPLS based IP networks. The Fifth IEEEISCC. – Antibes, France, 2000. – P. 214–219.
2. Погорілий С.Д., Білоус Р.В. Генетичний алгоритм розв'язання задачі маршрутизації в мережах // Проблеми програмування. – 2010. – № 2–3. – С. 171–178.
3. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. – СПб.: Питер, 2001. – С. 572–597.
4. Lee Y., Seok Y., Choi Y. A constrained multipath Traffic engineering scheme for MPLS networks. ICC'02 – New York: Red Hook, 2002. – P. 2431–2436.
5. Goldberg D.E. Genetic Algorithms in Search, Optimization and machine Learning. Reading, MA: Addison-Wesley, 1989. – P. 57–88.
6. Srinivas M., Patnaink L. Adaptive probabilities of crossover and mutation in genetic algorithms // IEEE Transactions on Systems, MAN and Cybernetics. – 1994. – 24 (4). – P. 656–667.

Поступила 01.04.2011

Тел. для справок: (044) 526-0522, (066) 287-3100 (Киев)

E-mail: sdp@univ.kiev.ua, romanrpd@gmail.com

© С.Д. Погорельый, Р.В. Білоус, 2012