МЕТОДИКА ОПТИМИЗАЦИИ СЕТКИ ЛИНИЙ СЕТИ СВЯЗИ КАК ПРОГРАММНО-КОНФИГУРИРУЕМОЙ СЕТИ

С.А. Шинкарев, *к.т.н., доцент, Военная академия связи им. Маршала Советского Союза С.М. Буденного, se_men82@mail.ru;*
Н.В. Евглевская, *к.т.н., Военная академия связи им. Маршала Советского Союза С.М. Буденного, n.evglevskaya@gmail.com;*
П.Н. Антонов, *Военная академия связи им. Маршала Советского Союза С.М. Буденного, maloy-ozr@mail.ru.a;*
Т.Р. Магдеев, *Военная академия связи им. Маршала Советского Союза С.М. Буденного, timazel@mail.ru.*

УДК 004.716

Аннотация. Решение задачи оптимизации обуславливается формированием структуры сети связи, удовлетворяющей определенным требованиям. В данной статье предложена методика и алгоритм построения линий связи на заданной сети.

Ключевые слова: программно-конфигурируемая сеть; задача коммивояжера; алгоритм; узел; сетка линий.

METHODOLOGY FOR LINE GRID OPTIMIZATION OF NETWORK AS A SOFTWARE-DEFINED NETWORKING

Semen Shinkarev, *candidate of Engineering, assistant professor, The Military Academy of Telecommunications named after Marshal of the Soviet Union S.M. Budyonny;*
Natalya Evglevskaya, *candidate of Engineering, The Military Academy of Telecommunications named after Marshal of the Soviet Union S.M. Budyonny;*
Pavel Antonov, *The Military Academy of Telecommunications named after Marshal of the Soviet Union S.M. Budyonny;*
Timur Magdeev, *The Military Academy of Telecommunications named after Marshal of the Soviet Union S.M. Budyonny.*

Annotation. Optimization task solution is determined by the formation of a network structure that satisfies certain requirements. This article proposes a methodology and algorithm for constructing communication lines on a given network.

Keywords*:* software-defined networking; travelling salesman problem; algorithm; node; line grid.

Введение

В настоящее время состояние телекоммуникационных сетей показывает, что возможности базовых сетевых технологий, включающие в себя согласованный набор протоколов и аппаратно-программных средств, близки к исчерпанию. Это приводит к повышению загруженности линий сети связи, вследствие чего увеличивается время передачи информации. Одним из вариантов решения этой проблемы является переход на концепцию программно-конфигурируемых сетей. В последние годы данный вопрос рассматривается в научных работах, авторы которых предлагают использовать метод оптимизации сетки линий сети связи для повышения надежности сети.

Методика оптимизации линий сети связи

После определения количества и местоположения сетевых узлов, необходимо сформировать сетку линий таким образом, чтобы полученный вариант структуры удовлетворял предъявляемым требованиям.

Данное построение является задачей коммивояжера [1-3]. В общем виде задача коммивояжера носит комбинаторный характер.

Пусть $P = 1, i_2, i_3, ..., i_N$ – некоторая перестановка чисел $1, 2, ..., N$, определяемая количеством узлов структуры сети:

$$P = \begin{pmatrix} 1 & 2 ... N \\ 1 & i_2 ... \ i_N \end{pmatrix}. \tag{1}$$

Пусть известна общая матрица расстояний на сети $L = \|l_{ij}\|$ вида $N \times N$.

Если в множестве P_N всех перестановок вида (1) выделить множество всех полных циклов, то задачу коммивояжера можно представить в следующем виде.

Каждой перестановке (1) сопоставим выражение (2), которое называется длиной перестановки (1):

$$\mathbf{C}_l(p) = \sum_{i=1}^{N} l_{i\ p(i)}\ . \tag{2}$$

Таким образом, решение задачи коммивояжера заключается в нахождении из P_N перестановки (цикла) p_0, для которой $C_l^*(p_0)$ минимально на множестве длин (2), вычисленных для всех полных циклов, определяемых $C_l^*(p_0) = \arg \min_{p\hat{I} P_N} \{C_l(p)\}$.

Задача коммивояжера является трудно решаемой задачей комбинаторной оптимизации.

Число алгоритмов решения этой задачи как точных, так и приближенных превышает несколько десятков [1-3]. Наиболее известным из существующих является алгоритм Литтла, основанный на методе ветвей и границ, обладающий практически экспоненциальной сложностью. Недостатком данного алгоритма является то, что он способен решать задачи при небольшом количестве узлов, а также относительно сложен для программной реализации.

Наиболее приемлемыми и позволяющими решать задачи с реальной размерностью сети, являются два метода:

1) метод динамического программирования для задач упорядочивания, предложенный М. Хелдом и Р. Карпом [4-6];

2) метод Р. Беллмана применительно к задаче о коммивояжере [7-9].

Данные методы были предложены независимо.

Метод Беллмана рассматривается как многошаговый процесс принятия решения. Путь является замкнутым, но начальным является узел 0. На некотором оптимальном пути узел i становится достигнутым и перед возвращением в узел 0 остается посетить k узлов $j_1, j_2, ..., j_k$. В связи с тем, что путь является оптимальным, его часть, соединяющая в некотором порядке узлы $j_1, j_2, ..., j_k$, должна иметь минимальную длину, в ином случае путь не был бы оптимальным. Уравнение, выражающее принцип оптимальности, представлено выражением $f(i; j_1, j_2, ..., j_k) = \min_{1 \le m \le k} \{l_{ij_m} + f(i; j_1, j_2, ..., j_{m-1}, j_{m+1}, ..., j_k)\}$, рекурсивная функция:

$$f(i;j) = l_{ij} + l_{j0}, \qquad (3)$$

из которой можно найти $f(i; j_1, j_2)$. С помощью уравнения (3) можно определить $f(i; j_1, j_2, j_3)$ и т.д. до тех пор, пока не будет вычислено $f(0; j_1,..., j_n)$, где l_{ij} – расстояние между узлами i и j, а последовательность m дает искомый минимальный путь.

Метод Беллмана является простым для реализации и имеет небольшую вычислительную сложность при небольшом количестве узлов. При значительном увеличении количества узлов решение задачи методом Беллмана в режиме реального времени не представляется возможным вследствие экспоненциальной сложности.

Для получения варианта решения в режиме реального времени необходимо введение обязательной эвристической процедуры, которая позволила бы разбить перестановку для нахождения не всего цикла, а лишь его фрагментов. Метод Беллмана не позволяет декомпозировать задачу, так как рассматриваемый путь в задаче является замкнутым.

Метод Хелда и Карпа дает последовательность перестановок, каждая из которых получается из предыдущей путем решения некоторой подзадачи сравнительно небольшого объема с той же структурой, что и данная, т.е. с помощью метода Хелда и Карпа можно декомпозировать задачу. Сущность метода заключается в следующем.

Пусть $S \subseteq \{2, 3, ..., N\}$ – некоторое подмножество узлов, $C(S, a_r)$ – путь минимальной длины для элементарной цепи, начинающийся в вершине 1, проходящий через все вершины из S и заканчивающийся в вершине $a_r \in S$.

Тогда:

$$(n(S) = 1) : C(\{a_r\}, a_r) = l_{1r}, \text{ для любого } a_r, \qquad (4а)$$

где: вершина a_m непосредственно предшествует a_r и все остальные вершины просматриваются в оптимальном порядке, получаем:

$$(n(S) > 1) : C(S, a_r) = \min_{a_m \in S \setminus a_r} [C(S - a_r, a_m) + l_{a_m a_r}]. \qquad (4б)$$

Если через C^* обозначить путь минимальной длины, включая возвращение в 1, то:

$$C^* = \min_{a_r \in \{2,3,...,n\}} [C(\{2,3,...,n\}, a_r) + l_{a_r 1}]. \qquad (5)$$

Перестановка $P = 1, a_2, a_3, ..., a_n$ оптимальна тогда и только тогда, когда:

$$C^* = C(\{2,3,...,n\}, a_r) + l_{a_r 1}, \qquad (6)$$

для $2 \le p \le n-1$

$$C(\{2,3,..., p, a_{p+1}\} + a_{p+1}) = C(\{2,3,..., p\}, a_p) + l_{a_p a_{p+1}}. \qquad (7)$$

Для получения оптимального решения вычисления проводятся в два этапа.

На первом этапе, используя (4б), необходимо определить $C(S, a_r)$. C^* определяется по формуле (5).

На втором этапе для нахождения оптимальной перестановки используются выражения (6) и (7).

Данный метод позволяет находить кратчайшую цепь не на всем множестве перестановок, а лишь на его части.

Задача разбиения перестановок P, представляющая собой путь через N узлов, на k упорядоченных подмножествах, каждое из которых состоит из городов, встречающихся последовательно в соответствии с перестановкой P и сохраняющих тот же порядок, что и в P, можно представить выражением:

$$P = (1, a_2, a_3, ..., a_n) = \bigcup_{u=1}^{k} S_u, \textit{ где } S_u \ \acute{I} \ \{a_2, a_3, ..., a_r, a_{r+1}\}.$$

Затем решается задача коммивояжера для k вершин, в котором каждое упорядоченное множество рассматривается как один город, а стоимость перехода определятся в соответствии с выражением:

$$S_u(a_q, a_{q+1}, ..., a_{m-1}, a_m) \rightarrow S_{u+1}(a_i, a_{i+1}, ..., a_{r-1}, a_r) = l_{a_m a_r}.$$

На рис. 1 представлена процедура разбиения на подмножества и сам метод нахождения цикла Гамильтона.

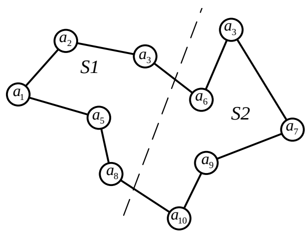

Рисунок 1

Следующий шаг формирования сетки линий – введение дополнительных ребер $b^*_{pg} \in \{b_{kl}\}$, где $n^* = |\{b_{kl}\}|$ – множество всех возможных ребер. Ребра должны вводиться в соответствии с условием:

$$b^*_{pg} \Rightarrow \max_{\{b^{(m)}_{pg} \notin \Gamma\}} \Delta N_{ОД},\ p, g = \overline{1, N},\ \ p \neq g,$$

согласно которому необходимо вводить только те ребра, которые дают максимальное количество остовных деревьев.

Процедура введения ребер продолжается до тех пор, пока ранг всех узлов не будет равен требуемому значению.

Количество ребер, которое может быть введено, определяется в соответствии с выражением: $n^* =]\frac{N\, r(a_i)}{2}[- N$, то есть процедура введения ребер конечна.

Дополнительные ребра вводятся с использованием метода разложения графа по ребру [10-12]. Данный метод направлен на увеличение $N_{\text{од}}$ – количества остовых деревьев и уменьшение диаметра и среднего расстояния на сети. Суть метода заключается во введении дополнительных ребер, имеющих максимальный вес в матрице весов ребер D:

$$D^{(m)} = \left\| \Delta N_{\text{ОД}}^{pg} \right\|,$$

$$\Delta N_{\text{ОД}}^{pg\,(m)} = \begin{cases} \Delta N_{\text{ОД}}^{pg},\ p, g = \overline{1, N},\ p \neq g \\ 0,\ p = g, \end{cases},$$

элементы этой матрицы определяются согласно выражению:

$$\Delta N_{\text{ОД}}^{(m)} = N_{\text{ОД}}^{(m)}(\{\Gamma_i\} \bigcup [\{b_{kl}\}^{(m-1)} \bigcup b_{pg} \notin \{\Gamma_i\}]) - N_{\text{ОД}}^{(m-1)}(\{\Gamma_i\} \bigcup [\{b_{kl}\}^{(m-1)} \notin \{\Gamma_i\}]),$$

где количество остовных деревьев в топологической структуре подсистемы связи и передачи данных: $G(A, B) = \Gamma \cup b_{kl}$, где $b_{kl} \notin \Gamma$.

На рис. 2 представлен пример введения дополнительных ребер.

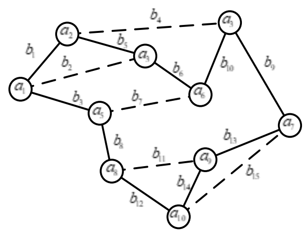

Рисунок 2

После введения дополнительных ребер производится трансформация сети [13-15] с целью увеличения ранга минимальных сечений и выравнивания основной системы сечений.

Для этого необходимо найти в графе сечение минимального ранга:

$$G(A, B_e, \overline{A}),\ B_e = \min_{\sigma_{ij}^e}\{b_{ij}\} = \sigma_{ij}^{\min},$$

разбить граф на множества A и $\overline{A}$ $G_{s-1} : [A, \overline{A}]$, менять ребра «крест на крест», при этом брать их из каждой вершины попарно:

$$\begin{cases} b_a = (a, a'); \\ b_{\overline{a}} = (\overline{a}, \overline{a}'),\ a, a' \in A; \\ \overline{a}, \overline{a}' \in A,\ b_a, b_{\overline{a}} \in \Gamma_i; \\ G_s = G_{s-1} \bigcup \{b_c, b_d\} \setminus \{b_a, b_{\overline{a}}\}; \\ b_c = (a, \overline{a}'),\ b_d = (a', \overline{a}), \end{cases}$$

до тех пор, пока не будет получена структура с выровненными сечениями и рассекающими множествами на всей сети, ранг которых будет равен или отличаться на единицу $r(\sigma_1) \approx r(\sigma_2) \approx ... \approx r(\sigma_n)$.

Осуществим трансформацию сети на примере, представленном на рис. 2 «Процесс введения дополнительных ребер».

Разобьем узлы на два подмножества сечением $\sigma_{\min} = \{b_8, b_9\}$ $A = \{a_1, ..., a_6\}$ и $\overline{A} = \{a_7, ..., a_{10}\}$.

Выберем произвольное ребро в сечении $[A, \bar{A}]$, например $b(a_5, a_8)$. Существует по крайне мере два ребра, соединяющих узел a_5 с остальными узлами в A. Из этих узлов один a_1 не смежен с узлами в $\bar{A}$. Подобно этому, в $\bar{A}$ существует хотя бы один узел, например a_9, смежный с узлом a_8, но не смежный ни с одним узлом в A. Трансформация состоит в удалении $b(a_5, a_1)$ и $b(a_8, a_9)$ из G и добавления $b(a_1, a_8)$ и $b(a_5, a_9)$, как показано на рис. 3, где раскрыт процесс трансформации сети (пример 1).

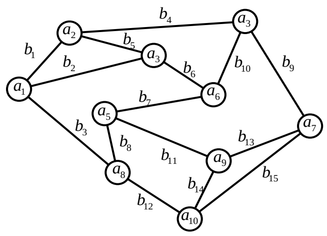

Рисунок 3

Таким образом, трансформация не влияет на ранг узлов, однако позволяет уменьшить число сечений минимального ранга.

Реализуя дальнейшую трансформацию сети, должны получить сечения с одинаковым рангом или отличным на единицу.

Найдем минимальное сечение в графе (рис. 3) $\sigma_{\min} = \{b_3, b_7, b_9\}$. Определим узлы из подмножеств, не инцидентных узлам подмножеств a_3 и a_9, с соответствующими ребрами $b(a_3, a_6)$ и $b(a_5, a_9)$. Произведем трансформацию, результаты которой представлены на рис. 4, где раскрыт процесс трансформации сети (пример 2).

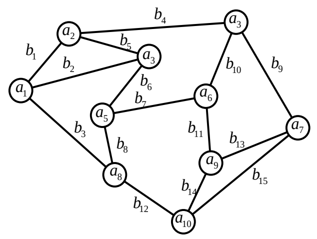

Рисунок 4

Рассмотрим следующее минимальное сечение $\sigma_{\min} = \{b_4, b_7, b_{12}\}$. Узлы a_8 и a_7 с ребрами $b(a_5, a_8)$ и $b(a_7, a_9)$. Трансформация представлена на рис. 5, где представлен процесс трансформации сети (пример 3).

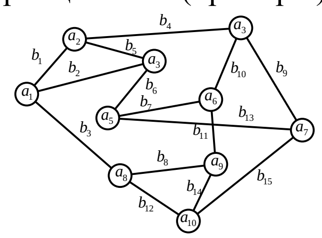

Рисунок 5

Если дальше производить трансформацию сети, то можно получить граф $G'(A,B')$, представленный на рис. 6, рассматриваемый как вариант топологической структуры. По сравнению с первоначальным вариантом $G(A,B)$ (рис. 3), в данном графе сечения выровнены и имеют более высокий ранг, в соответствии с условием.

В итоге получена топологическая структура (рис. 6), обладающая рядом характеристик, имеющих экстремальные значения: максимальное количество остовых деревьев; минимальный диаметр и среднее расстояние; требуемая узловая и вершинная связности; мощности сечений основной системы, которые максимизированы и выровнены; сетевые узлы, расположенные на необходимом расстоянии или близком к нему; требуемый ранг вершин.

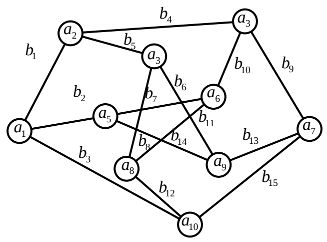

Рисунок 6

Оптимальные значения указанных теоретико-графовых характеристик достигаются именно на неиерархических квазиоднородных структурах при рациональном расходе ресурса сил и средств на развертывание сети связи.

Заключение

В статье представлена методика формирования сетки линий сети связи, позволяющая получить структуру сети с повышенной надежностью и производительностью.

Литература

1. Меламед И.И., Сергеев С.И., Сигал И.Х. Задача коммивояжера. Точные алгоритмы // Автоматика и телемеханика, 1989. – № 10. – С. 3-29.
2. Панюков А.В., Леонова Ю.Ф. Алгоритм соединения циклов для метрической задачи коммивояжера на максимум // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика, 2021. – № 4. – С. 26-36.
3. Борисова Е.С., Мельников Б.Ф. Аппроксимационные алгоритмы и псевдометрический вариант задачи коммивояжера // Известия высших учебных заведений. Поволжский регион. Физико-математические науки, 2009. – № 3. – С. 96-100.
4. Хелд М., Карп Р.М. Применение динамического программирования к задачам упорядочивания // Кибернетический сборник, 1964. – № 9. – С. 182-218.
5. Сесекин А.Н., Ченцов А.А., Ченцов А.Г. Задачи маршрутизации перемещений. Учебное пособие для вузов. – СПб.: М.: Краснодар.: Изд-во Лань, 2022. – 240 с.
6. Саттон Р.С., Барто Э. Дж. Обучение с подкреплением: введение. – М.: Изд-во ДМК Пресс, 2020. – 552 с.
7. Беллман Р. Применение динамического программирования к задаче о коммивояжере // Кибернетический сборник, 1964. – № 9. – С. 219-222.
8. Окулов С.М., Пестов О.А. Динамическое программирование. – М.: Изд-во «Лаборатория знаний», 2020. – 299 с.

9. Ченцов А.Г. Задача маршрутизации «На узкие места» с системой первоочередных заданий // Известия Института математики и информатики Удмуртского государственного университета, 2023. – № 1 (61). – С. 156-186.
10. Фрэнк Г., Фриш И. Сети, связь и потоки. – М.: Изд-во Связь, 1978. – 448 с.
11. Велигоша А.В. Общая теория связи: учебное пособие. – Ставрополь.: Изд-во СКФУ, 2014. – 240 с.
12. Берлин А.Н. Высокоскоростные сети связи: учебное пособие. – М.: Изд-во ИНТУИТ, 2016. – 451 с.
13. Давыдов Г.Б. Сети электросвязи. – М.: Изд-во «Связь», 1977. – 360 с.
14. Цветков Ф.А. Программно-конфигурируемые радиоустройства: принципы построения и алгоритмы обработки сигналов: учебное пособие. – Ростов-на-Дону.: Изд-во ЮФУ, 2020. – 163 с.
15. Гольдштейн Б.С., Елагин В.С., Зарубин А.А., Селиванов А.Е. Программно-конфигурируемые сети SDN. Протокол OPENFLOW: учебное пособие. – Санкт-Петербург.: Изд-во СПбГУТ им. проф. М.А. Бонч-Бруевича, 2018. – 47 с.