

УДК 004.023:629.3.072.1-022.233

РАЗРАБОТКА ЭВРИСТИЧЕСКОГО АЛГОРИТМА ПОИСКА ОПТИМАЛЬНОГО МАРШРУТА



А.А. Навроцкий

Заведующий кафедрой информационных технологий автоматизированных систем БГУИР, кандидат физико-математических наук, доцент



Р.В. Козарь

Магистрант кафедры информационных технологий автоматизированных систем БГУИР

*Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: navrotsky@bsuir.by, pozitr0n.kozarroman@gmail.com*

Аннотация. Рассматриваются вопросы разработки нового эвристического алгоритма.

Ключевые слова: алгоритм, эвристика, метаэвристика, маршрут, окрестность, локальный поиск

В настоящее время практически на всех предприятиях необходимо решать вопросы, связанные с транспортной логистикой. Одной из проблем является необходимость организации своевременной доставки товаров с учетом нестабильной дорожной ситуации. Для решения этой проблемы в программные продукты автоматизации транспортной логистики внедряются алгоритмы, позволяющие строить оптимальный маршрут доставки товаров при различных дорожных ситуациях. Наличие большого числа транспортных средств повышает требования к качеству таких программ.

В данном докладе представлена модификация эвристического алгоритма локального поиска для нахождения оптимального маршрута доставки товаров. Вначале работы алгоритма выбирается начальное решение (найденное любым алгоритмом или выбранное случайно), которое затем итерационно приводится к оптимальному решению.

На каждом шаге локального спуска происходит переход от текущего решения к соседнему решению с меньшим значением целевой функции до тех пор, пока не будет достигнут локальный оптимум [1].

Соседнее решение не обязательно должно быть наилучшим в окрестности, но критерий оценки решения не должен меняться во время итеративного процесса. Таким образом, для любого решения s должно быть задано некоторое множество $N(s)$ соседних решений, которое и называется окрестностью s .

Поведение алгоритма зависит от следующих факторов:

1 Структуры окрестностей N . Размер окрестности любого решения должен выбираться на основе компромисса между целью получения хорошего улучшения при каждом переходе к новому решению и целью ограничения времени просмотра одной окрестности. Обычно, для любого решения s , окрестность $N(s)$ порождается с помощью некоторой операции локального изменения s (применительно к текущей ситуации – это абсолютно непредсказуемое изменение дорожной обстановки).

2 Начального решения s_0 . Его можно находить с помощью любого алгоритма (например, конструктивной эвристики), который выдает допустимое решение, или с помощью процедуры случайной генерации значений.

3 Стратегии выбора новых решений. Например, просмотр всех решений из $N(s)$ и выбор наилучшего. Если найденное решение не является оптимальным для локальной области, то осуществляется переход к наилучшему соседу, или к первому лучшему решению, найденному в окрестности.

В эвристиках локального поиска используются окрестности, определяемые с помощью последовательностей ограниченной длины операций локального обмена [2]. Пусть, при фиксированном целом $k > 0$, решается этим методом индивидуальная задача $x \in I_p$ проблемы P .

Решение y находится в k -обменной окрестности $N^k(s)$, если из решения s можно получить решение y , применив не более чем k операций локального обмена. Эвристики, которые основаны на k -обменных окрестностях, часто называют k -оптимальными (k -opt) эвристиками.

Рассмотрим метод основанный на следующей 2-обменной окрестности: для любого тура τ 2-обменная окрестность $N(\tau)$ это множество всех туров τ' , которые могут быть получены из τ после удаления двух ребер (x, y) и (u, z) и добавления двух новых ребер (x, u) и (z, y) . Тот же процесс построения нового тура τ' из тура τ можно описать так: выделяется некоторая цепь, которая в новом туре проходит в противоположном направлении. Такая окрестность содержит $\frac{n(n-3)}{2}$ элементов, что несколько меньше, чем в окрестности city-swap.

Алгоритм 2-opt содержит в себе и реализует 2-оптимальную эвристику для задачи о коммивояжере и является примером локального поиска с 2-обменной окрестностью.

Представленный алгоритм построения оптимального маршрута является улучшенной модификацией эвристического алгоритма 2-opt локального поиска, в частности, локального спуска.

Алгоритм базируется на принципе эвристического локального поиска для задачи коммивояжера, однако в нем используется 3-обменная окрестность [3].

Введена функция l , которая определена на множестве туров следующим образом:

$$l(\tau) = \sum_{i=1}^{n-1} D(\pi(i), \pi(i+1)) + D(\pi(n), \pi(1)), \quad (1)$$

где: $\tau = \langle c_{\pi(1)}, \dots, c_{\pi(n)} \rangle$.

Так же необходимо отметить, что для задачи коммивояжера с n городами k -обменная окрестность имеет размер $\binom{n}{k} = \Theta(n^k)$. Следовательно, можно сделать вывод о том, что данному новому эвристическому алгоритму потребуется выполнить $O(n^k)$ шагов для того, чтобы удостовериться в том, что текущее решение является эффективным и локально оптимальным.

Предложенная модификация алгоритма локального поиска имеет принципиально новую схему работы. Она заключается в том, что алгоритм не останавливается в точке локального оптимума, как это делается в алгоритме локального спуска 2-обменной окрестности, а продолжает работать для поиска точки, приближенной к глобальному оптимуму [4].

Был разработан механизм, который позволяющий осуществить выход из локального оптимума. Данный метод называется исключаящий список $List(i_k)$.

Данный список строится по предыстории поиска, то есть, по нескольким последним точкам $i_k, i_{k-1}, \dots, i_{k-l+1}$, и запрещает исследовать часть окрестности $N(i_k)$ текущего решения i_k . Точнее, на каждом шаге алгоритма очередная точка i_{k+1} является оптимальным решением следующей подзадачей, описанной формулой:

$$m(i_{k+1}) = \min\{m(j) | j \in N(i_k) \setminus List_l(i_k)\} \quad (2)$$

Множество $List_l(i_k) \subseteq N(i_k)$ определяется по предшествующим решениям. Данный список исключений учитывает специфику задачи и, как правило, запрещает использование тех «фрагментов» решения (ребер графа, координат вектора, цвет вершины), которые менялись на последних l шагах алгоритма. Константа $l \geq 0$ определяет его память.

При «короткой памяти», когда $l = 0$, получаем стандартный локальный спуск [5].

Работа алгоритма *3-opt*:

1. Выбор начального решения i_0 и положим, что $m^* = m(i_0), k = 0$.

2. Пока не выполнен критерий остановки выполнять следующее:

2.1. Сформировать окрестность $N_p(i_k)$.

2.2. Если $N_p(i_k) \neq \emptyset$, тогда $i_{k+1} := i_k$, иначе найти i_{k+1} такой, что: $m(i_{k+1}) = \min\{m(j) | j \in N(i_k) \setminus List_l(i_k)\}$.

2.3. Если $m^* > m(i_{k+1})$, то $m^* := m(i_{k+1})$.

2.4. Положить $k := k + 1$ и обновить список $List_l(i_k)$.

Проведен тест трех алгоритмов. В качестве среды для анализа был использован Matlab. В качестве функций для тестирования были взяты логарифмическая и линейная функции с различными коэффициентами:

- *1-opt* 1-окрестности;
- *2-opt* 2-окрестности;
- *3-opt* 3-окрестности (новый алгоритм).

Результаты первого теста представлены на рисунке 1.

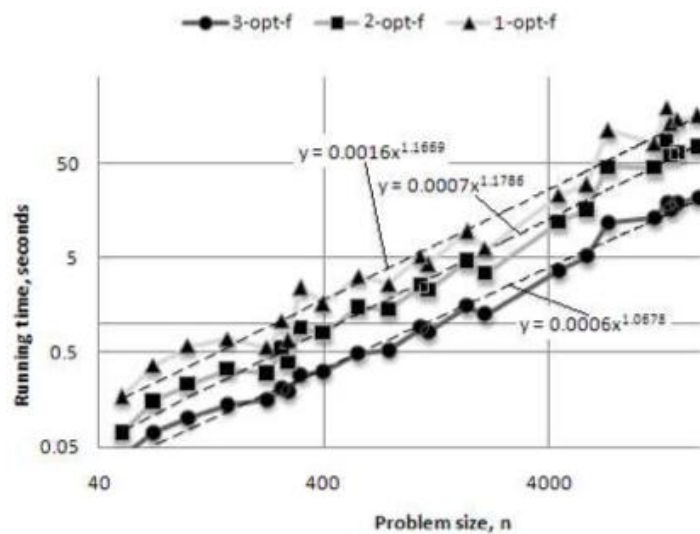


Рисунок 1. Результаты первого теста

Из графика видно, что производительность 1-окрестного алгоритма весьма низкая, поскольку его временные затраты очень высоки.

Данные 2-окрестного алгоритма несколько лучше, но также достаточно затратные. Данные 3-окрестного алгоритма намного лучше, чем 1-окрестный и 2-окрестный алгоритмы, что показывает шкала временных затрат.

Так же, для большей наглядности и достоверности эксперимента было проведено второе тестирование. Результаты второго тестирования алгоритмов поиска оптимального маршрута представлены на рисунке 2.

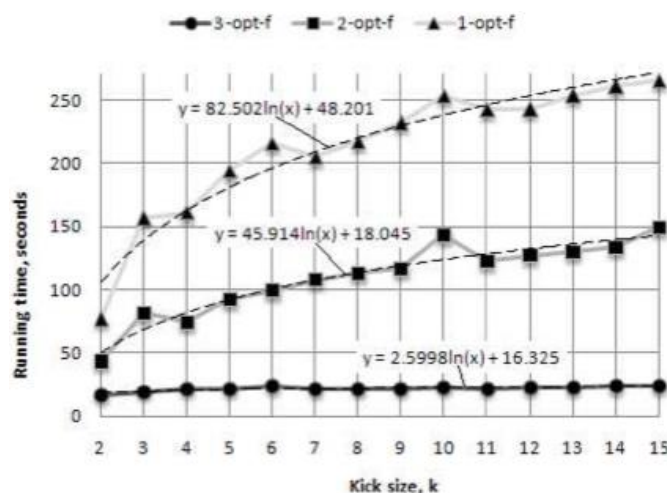


Рисунок 2. Результаты второго тестирования

На основании полученных данных, можно утверждать, что предложенный алгоритм 3-opt имеет более высокую производительность и более высокое качество результата.

Список литературы

[1]. Фирма 1С. Базовые алгоритмы нахождения кратчайших путей во взвешенных графах [<https://habrahabr.ru/post/119158/>] / Фирма 1С. – Электрон. текстовые дан. – Режим доступа <https://habrahabr.ru/company/1c/profile/>, свободный.

[2]. Фирма ArcGIS. Эвристические алгоритмы, используемые дополнительным модулем ArcGIS Network Analyst [<https://desktop.arcgis.com/ru/arcmap/10.4/extensions/network-analyst/algorithms-used-by-network-analyst.htm>] / Фирма ArcGIS. – Электрон. текстовые дан. – Режим доступа <https://desktop.arcgis.com/ru/>, свободный.

[3]. Фирма Websketches. Эвристические алгоритмы поиска кратчайшего пути в браузерных играх [<http://websketches.ru/blog/algorithm-poiska-pitu-v-brauzernih-igrah>] / Фирма Websketches. – Электрон. текстовые дан. – Режим доступа <http://websketches.ru/>, свободный.

[4]. Программные продукты и системы, Интеллектуально-экспертный метод определения оптимального маршрута транспортировки продукции [<http://www.swsys.ru/index.php?page=article&id=3690>] / Программные продукты и системы. – Электрон. текстовые дан. – Режим доступа <http://www.swsys.ru/>, свободный.

[5]. Фирма АБВУ, Локальный поиск в пространстве состояний [<https://habrahabr.ru/company/abbyu/blog/217839/>] Фирма АБВУ. – Электрон. текстовые дан. – Режим доступа <https://habrahabr.ru/company/abbyu/>, свободный.

DEVELOPMENT OF THE HEURISTIC ALGORITHM FOR SEARCHING THE OPTIMUM ROUTE

A.A. NAUROTSKY, PhD
Head of the Department of Information Technologies of Automated Systems of BSUIR, Associate Professor

R.V. KOZAR
Master of Information Technologies of Automated Systems of BSUIR

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
E-mail: navrotsky@bsuir.by, pozitron.kozarroman@gmail.com

Abstract. The questions of development of a new heuristic algorithm are considered.

Key words: algorithm, Heuristics, Meta-heuristics, Route, Neighborhood, Local search.