

МОДИФИКАЦИЯ ДВУХЭТАПНЫХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ МАРШРУТИЗАЦИИ С ВРЕМЕННЫМИ ОКНАМИ

Егорова О.Е., Закирова У.В., Осечкина Т.А.

ФГБОУ ВПО «Пермский национальный исследовательский политехнический университет», Пермь, Россия (614990, г. Пермь, Комсомольский пр., 29), egorovaolgagm@gmail.com

На сегодняшний день задача инкассации является одной из наиболее актуальных из класса задач маршрутизации, поскольку каждая компания, предоставляющая подобные финансовые услуги, нуждается в оптимизации своих затрат. Данная работа посвящена вопросу модификации двухэтапных алгоритмов для решения задач маршрутизации с временными окнами, поскольку данные задачи являются трудноразрешимыми. В работе приводится математическая модель с двумя дополнительными ограничениями: по времени и количеству человек, которое необходимо для обслуживания того или иного объекта. Показан алгоритм в виде блок-схемы с подробным описанием основной процедуры, а также приведен пример применения данного алгоритма. Рассматриваемая модификация позволяет создавать приближенное решение, которое в дальнейшем нуждается в доработке с помощью улучшающих методов. Представлена блок-схема процедуры по формированию маршрута.

Ключевые слова: задача инкассации, оптимальный маршрут, кластеризация, приближенное решение, Vehicle Routing Problem with Time Window, жадный метод.

MODIFICATION OF TWO-STAGE ALGORITHMS FOR THE SOLUTION OF PROBLEMS OF ROUTING WITH TEMPORARY WINDOWS

Egorova O.E., Zakirova U.V., Osechkina T.A.

Perm National Research Polytechnic University, Perm, Russia (614990, Perm, avenue Komsomol, 29), egorovaolgagm@gmail.com

Today the problem of collection is one of the most actual of a class of problems of routing as each company providing similar financial services, needs optimization of the expenses. This work is devoted to a question of modification of two-stage algorithms for the solution of problems of routing with temporary windows as these tasks are stubborn. The mathematical model is given in work with two additional restrictions: on time and a number of people which is necessary for service of this or that object. The algorithm in the form of the flowchart with the detailed description of the main procedure is shown, and also the example of application of this algorithm is given. Considered modification allows to create the approximate decision which needs further completion by means of improving methods. The procedure flowchart on route formation is submitted.

Keywords: problem of collection, optimum route, clustering, approximate decision, Vehicle Routing Problem with Time Window, greedy method.

На сегодняшний день задача инкассации является одной из наиболее актуальных из класса задач маршрутизации, поскольку каждая компания, предоставляющая подобные финансовые услуги, нуждается в оптимизации своих затрат.

Существует множество вариаций различных ограничений для данного типа задач, например, ограничения по времени заезда, по количеству депо и по виду транспортных средств, и применяются они как совместно, так и по отдельности [6].

Рассмотрим классическую задачу с дополнительными ограничениями: по времени и количеству человек, которое необходимо для обслуживания того или иного объекта. Для данной задачи математическая модель примет следующий вид:

$$F = \sum_{l=1}^m \sum_{\substack{i=1 \\ j=1 \\ i \neq j}}^k t_{ij}^l \rightarrow \min \quad (1)$$

$$\sum_{l=1}^m \sum_{i,j=1}^n x_{ij}^l = 1, \quad \sum_i x_{ip}^l - \sum_j x_{pj}^l = 0, \quad l = \overline{1, m}, \quad x_{ij}^l \in \{0; 1\} \quad (2)$$

$$\sum_{l=1}^m Q_l^+(x_{ij}) + \sum_{l=1}^m Q_l^-(x_{ij}) \leq q \quad (3)$$

$$l_j - a_j - e_o^l - \sum_{i=0}^{j-1} \sum_{p=1}^j t_{ip}^l \geq 0, \quad l = \overline{1, m} \quad (4)$$

$$r(x_{ij}) \leq r_{ij} \quad (5)$$

Целевая функция (1) минимизирует время обслуживания объекта, здесь t_{ij}^l – время, затрачиваемое на преодоление пути от объекта i до объекта j ; l – индекс, отвечающий за обход по маршрутам; m – количество маршрутов. Ограничение (2) означает, что каждый объект обслуживается один раз, а также может принадлежать только одному лепестку. Ограничение (3) подразумевает под собой максимальный объем груза, который вмещается в транспортное средство, где q – грузоподъемность автомобиля; Q^+ – вес точки, который необходимо погрузить в машину; Q^- – вес, который необходимо выгрузить. (4) – ограничение на время закрытия объекта: l_j – время закрытия объекта, a_j – время обслуживания точки, e_o^l – время выезда l -ой бригады из депо. Ограничение (5) говорит о том, что количество обслуживающих рабочих, приехавших на объект, не должно быть меньше, чем необходимо данному объекту.

Данная задача относится к NP-трудных задач. Существует несколько методов решения таких задач, а именно – точные, эвристические и метаэвристические методы [4]. Точные методы основываются на полном переборе всех возможных решений, что, в свою очередь, делает их неэффективными, поскольку длительность поиска решений занимает большой промежуток времени [2]. Минус точных методов учитывается в эвристических. Они производят относительно ограниченный поиск решений, которые дает довольно хорошее решение за приемлемое время, но решение является приближительным [2]. Самыми оптимальными методами, которые находят довольно быстро и хорошее решение, являются метаэвристические методы. Но в данных методах также имеется минус – есть параметр, который напрямую влияет на результат, исходя из входных данных [3].

Метаэвристические методы представляют собой двухэтапные алгоритмы [7]. Идея данных методов заключается в следующем: на первом этапе находится приближенное решение, на втором происходит его улучшение. При реализации второго этапа сложностей не возникает, поскольку улучшающие методы хорошо изучены, к ним можно отнести метод отжига, генетический алгоритм и муравьиный алгоритм и др. Но была обнаружена проблема нахождения метода, который на первом этапе получал бы маршрут наиболее близкий к оптимальному, т.е. с небольшой погрешностью.

Задача в данной постановке подробно разбирается в работе [1]. В ней рассмотрен подробно алгоритм решения выше поставленной задачи, а также приведена блок-схема изложенного метода.

В данной работе представлены доработанные процедуры по формированию маршрута: Sort By Last Time и AddPoint Without Waiting. Ниже приведена блок-схема новой функции AddPoint, которая включает в себя вышеуказанные процедуры (рис.1).

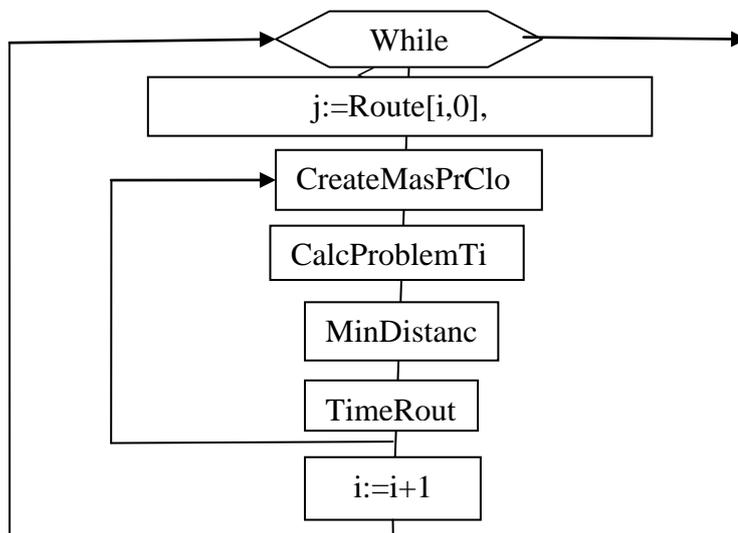


Рис. 1. Блок-схема процедуры AddPoint

Алгоритм, заложенный в основу данной процедуры, является модификацией жадного метода, т.е. в маршрут добавляется точка с минимальным расстоянием при условии, что она удовлетворяет всем ограничениям, наложенным на время.

В блоке CreateMasPrClose формируется массив, в котором точки расположены в порядке возрастания времени закрытия объектов при помощи формулы: $PrioritetClose[i] := l[i] - a[i] - TimeInWay - t[j^0, i]$, где j^0 – последняя включенная в маршрут. После формирования массива берется первый его элемент j^* и относительно него проводится процедура CalcProblemTime, которая показывает, успеет ли бригада обслужить другие объекты, если заедет в j^* объект. В основе данной процедуры лежит

следующая формула: $ProblemTime = PrioritetClose[i] - t[j^0, i] - a[j^*] + t[j^*, i]$. Если $ProblemTime$ для всех точек принимает значение больше нуля, то это означает, что бригада успеет во все точки, с учетом заезда в j^* объект.

В MinDistance формируется массив точек, которые находятся на минимальном расстоянии от точки j^* . Именно в данной процедуре заключается идея «жадного алгоритма» [5], т.е. бригада заезжает в ближние точки, в которые успеет. После того, как выбирается какой-либо объект k^* из массива MinDistance, проверяется условие на включение в маршрут данной точки по формуле: $PrioritetClose[j^*] = TimeRoute[i] + t[j^0, i]$, где $TimeRoute[i] = t[j^0, i] + t[i, j^*] + a[i]$. Если данная величина больше нуля, то ждать открытия объекта не нужно, и в данный объект можно заехать, прежде чем ехать в первый закрывающийся. После того, как объект включается в маршрут, необходимо поменять время бригады в пути.

Алгоритм выполняется до тех пор, пока все объекты не будут включены в маршрут. И данный набор процедур выполняется для всех лепестков.

Для проверки алгоритма был разработан программный продукт. Данные в программу скачиваются из файла MicrosoftExcel и хранятся в ней в виде динамических массивов. Это вызывает ряд неудобств, связанных с обращением к элементам массива, но делает программу более универсальной и оптимальной по количеству занимаемой памяти. Для хранения нескольких параметров, относящихся к одной точке, в ячейках массива используют записи с несколькими полями. Такой тип хранения переменных выбран в связи с тем, что, обращаясь к определенной переменной, можно быть уверенным в соответствии данных выбранной точке.

В работе берется 81 точка с учетом депо, то есть 80 объектов, которые необходимо обслужить. Количество человек, обслуживающих каждую из точек, составляет 2 или 3 человека. Время выезда бригад из депо одинаково, также как и вместимость транспортных средств. Заданы различные веса для каждой точки, различное время закрытия и открытия, а также время обслуживания.

После запуска программы получено 6 лепестков. Рассмотрим более подробно 4 лепесток. На рис.2 представлен маршрут движения транспортного средства по выбранному лепестку.

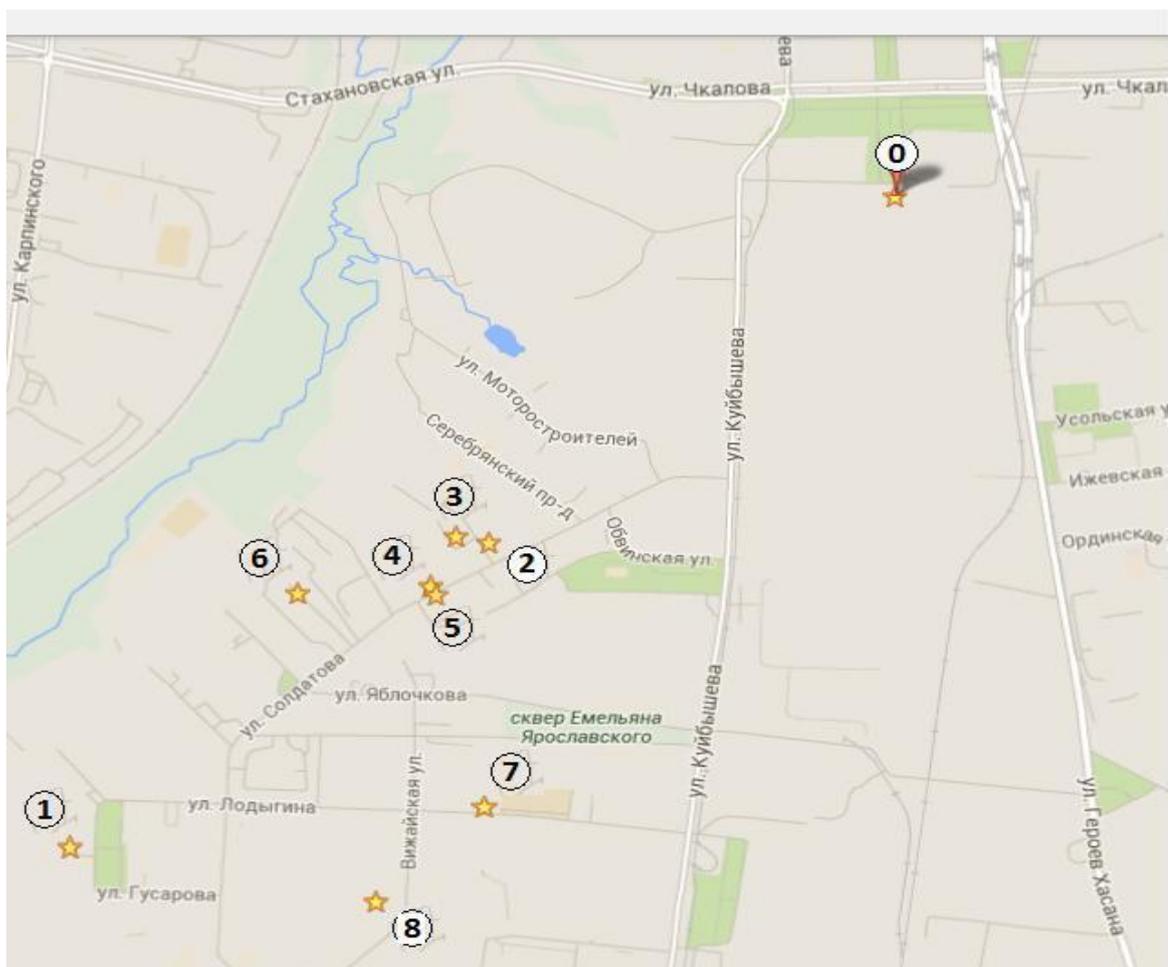


Рис. 2. Маршрут 4 лепестка

Маршрут по лепестку проложен оптимально, кроме переезда из точки 1 в точку 2, так как расстояние между ними больше, чем расстояние от объекта 1 до объекта 6. Такой «скачок» обоснован тем, что пункт 2 открывается раньше, чем пункт 6.

Погрешность получившегося решения составляет 28 % по сравнению с теоретически рассчитанным маршрутом. Разница в результатах обосновывается тем, что алгоритм не содержит в себе способов оптимизации работы. Так как в случае, когда вводится ограничение на время, оптимальный план ухудшается, то полученную таким образом погрешность можно считать вполне приемлемой.

Минусом данного алгоритма также является то, что он адаптирован для точек, сконцентрированных на небольшой территории. В коде программы адреса хранения переменных из внешнего файла заданы статически, это может вызвать ряд неудобств, связанных с перекачиванием данных.

Среди положительных особенностей алгоритма стоит выделить учет несимметричности матрицы расстояний; быструю обработку большого объема данных; универсальность, за счет возможности менять входные данные (осуществлять скачивание из других файлов Microsoft Excel); уничтожение обработанных данных из оперативной памяти,

после закрытия программы; возможность посмотреть файл с полученными результатами, формирующийся в процессе выполнения алгоритма программой.

Проанализировав полученные результаты, можно сказать, что полученное решение адаптировано для реальных условий, так как затраты транспортных ресурсов являются адекватными; маршруты строятся с учетом выбранных ограничений; имеет место быстрое действие программного продукта; но в связи с выявленной погрешностью алгоритм нуждается в доработке с помощью улучшающих методов.

Список литературы

1. Закирова У.В., Осечкина Т.А. Модификации сбалансированного алгоритма решения задач инкассации с дополнительными ограничениями // Наука и бизнес: пути развития. – 2014. – № 3. – С.43-45.
2. Эйрих С.Н. Обзор методов решения задач маршрутизации транспорта // NEW MAGENTA PAPERS. – 2012. – № 1. – С. 22–29.
3. Gendreau M. Metaheuristics for the vehicle routing problem / M. Gendreau, G. Laporte, J.- Y. Potvin // Technical Report CRT-963, Centre de Recherchesur les Transports. Universit de Montral, jan 1994.
4. Laporte G. Classical Heuristics for the Vehicle Routing Problem / G. Laporte, F. Semet // Les Cahiers du GERAD, G98-54, Group for Research in Decision Analysis. Montreal, Canada, 1998.
5. Lawler E.L., Lenstra J.K., RinnooyKan A.H.G., Shmoys D.B. Sequencing and Scheduling: Algorithms and Complexity // Logistics of Production and Inventory. Handbooks in Operations Research and Management Science. Vol. 4. North-Holland: Elsevier, 1993, P. 445–522.
6. Review of different types of a problem of routing of transport // NEW MAGENTA PAPERS, 2012 №1, p. 11–19.
7. Sam R. Thangiah. A Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Heuristic for Vehicle Routing Problems with Time Windows // Practical Handbook of GENETIC ALGORITHMS: Complex Coding Systems, Volume III, Chapter 9, 1999.

Рецензенты:

Абдулаев А.Р., д.ф.-м.н., профессор, ФГБОУ ВПО «Пермский национальный исследовательский политехнический университет», г. Пермь.

Цаплин А.И., д.т.н., профессор, ФГБОУ ВПО «Пермский национальный исследовательский политехнический университет», г. Пермь.