

ЭВРИСТИЧЕСКИЕ И МЕТАЭВРИСТИЧЕСКИЕ МЕТОДЫ РЕШЕНИЯ ДИНАМИЧЕСКОЙ ТРАНСПОРТНОЙ ЗАДАЧИ

Т.С. Емельянова

Таганрогский технологический институт Южного федерального университета e-mail:

emelyanova_ts@mail333.com

Источник: <http://pitis.tsure.ru/files31/05.pdf>

В данной работе дана постановка динамической транспортной задачи с ограничением по времени. Проведен обзор и анализ методов решения задач данного типа. Особое внимание уделено методам поиска с запретами.

1. Введение

Данная статья посвящена динамическим транспортным задачам (ТЗ) с ограничениями по времени (DVRPTW – dynamic vehicle routing problems with time windows) и методам их решения [1]. Транспортные задачи или задачи маршрутизации транспортных средств возникают в различных областях деятельности человека: доставка товаров от поставщика к клиенту, доставка сырья на производство, сбор промышленных отходов, почтовая доставка и т. д. Так как цена перевозки различного рода товаров явно или не явно присутствует в их стоимости, то сокращение транспортных расходов является важной и насущной экономической задачей. Целью решения всех ТЗ является составление маршрутов транспортных средств минимальных по ценовым затратам. ТЗ с ограничением по времени являются подклассом ТЗ, в них учитывается время, в течение которого должен быть обслужен клиент. Являясь более сложными по постановке, данные задачи более полно описывают реальный процесс, т. к. во многих практических задачах доставки товаров время прибытия к клиенту и время обслуживания клиента играют существенную роль. К данным задачам относятся задачи доставки людей, почты, школьников, сбор промышленных отходов и пр. В транспортных задачах с ограничением по времени каждому клиенту соответствует промежуток времени, в течение которого клиент должен быть обслужен. Если все запросы клиентов известны заранее и неизменны, время движения от клиента к клиенту известно и тоже не изменяется, то такие задачи называются статическими ТЗ. Однако на практике запросы клиентов могут меняться во время осуществления транспортного плана, время движения из-за поломок или аварий тоже меняется, поэтому возникает новый класс задач динамические ТЗ с ограничением по времени. Данный класс задач более полно моделирует ТЗ, возникающие на практике, и поэтому позволяет найти лучшее решение по сравнению с менее адекватными моделями [1-2].

В данной статье будет дана математическая формулировка статической транспортной задачи с ограничением по времени. Затем будут рассмотрены изменяющиеся параметры ТЗ, которые появляются при переходе от статической версии ТЗ с ограничением по времени к динамической, далее эти параметры будем называть динамическими элементами ТЗ. И проведен обзор методов решения ТЗ с ограничением по времени от классических эвристических до метаэвристических методов, и сравнение данных методов. Особое внимание в данной статье уделяется методам поиска с запретами.

2. Статическая транспортная задача с ограничением по времени

ТЗ с ограничением по времени относится к классу задач маршрутизации автотранспорта (VRP – Vehicle Routing Problem). Задачи данного типа можно описать

следующим образом. Имеется некоторое количество автотранспорта, один склад (депо) и некоторое количество клиентов. Для каждого транспортного средства требуется составить маршрут, на протяжении которого транспортное средство посещает ряд клиентов (например, с целью доставки какого-либо груза). На маршрут каждого транспортного средства накладывается ряд ограничений. Каждый маршрут должен начинаться и заканчиваться в депо. Общее количество товаров, требуемых для доставки клиентам на данном маршруте данного транспортного средства не должно превышать его грузоподъемность. Каждый клиент обслуживается лишь одним транспортным средством и лишь единожды, т. е. не допускается посещение одного клиента двумя и более транспортными средствами. Каждый клиент должен быть обслужен в определенный промежуток времени, этот промежуток определяется двумя значениями, первое значение определяет время прибытия транспортного средства к клиенту, второе время отправления. Для данной задачи формулируются следующие цели (целевые функции): первичная цель – минимизировать общее количество транспортных средств, необходимых для обслуживания всех клиентов; вторичные – минимизировать общее время обслуживания всех клиентов и общее расстояние, пройденное всеми транспортными средствами [2].

В свою очередь ТЗ с ограничением по времени имеет несколько разновидностей по количеству используемого автотранспорта. Данная задача может расширяться, если учесть тот факт, что клиенты могут не только получать, но и возвращать товар – задача маршрутизации с возвратом и доставкой товара с ограничением по времени (PDPTW – Pickup and Delivery Problem with Time Windows) [1]. Может использоваться одно транспортное средство (single-vehicle PDPTW) или несколько транспортных средств. Далее рассмотрим классическую постановку ТЗ с ограничением по времени для нескольких транспортных средств. Согласно [2] – [4] математически ТЗ с ограничением по времени можно представить в виде графа $G = (N, A)$, где N – множество вершин, которые соответствуют набору клиентов (customers) и обозначаются $1, 2, \dots, n$, и вершины 0 и $n + 1$ соответствуют исходному депо (depot) из которого начинают и в котором заканчивают свой маршрут все автомобили; A – набор дуг соединяющих соответствующие вершины графа (соответствующих клиентов), если i – один клиент, а j – другой, то дуга их соединяющая обозначается $(i, j) \in A$. Обозначим C – множество клиентов $|C| = n$. Каждый клиент характеризуется определенным спросом $d_i, i \in C$. Каждой дуге $(i, j) \in A$ соответствуют время

t_{ij} – время перемещения от клиента $i, i \in C$ к клиенту $j, j \in C$, это время включает в себя время обслуживание клиента i и c_{ij} – стоимость пути автомобиля из i в j . Верхним индексом k , далее будет обозначаться соответствующий автомобиль (где $k \in V, V$ – количество идентичных автомобилей грузоподъемностью q). Каждый клиент должен быть обслужен в определенный промежуток времени, так называемое “временное окно” (time window), обозначаемый $[a_i, b_i], i \in C$. Время прибытия соответствующего автомобиля в определенную вершину графа обозначается S_i^k для $0 \leq i \leq n, \forall k \in V$. Время отправления из депо для всех автомобилей равно 0, т. е. $S_0^k = 0$ (для $\forall k \in V$). Переменные X_{ij}^k принимают значения $\{0, 1\}$, 1 означает, что автомобиль движется от вершины i к вершине j , 0 обратное. Согласно [2], графически транспортную задачу можно представить следующим образом см. рис. 1.

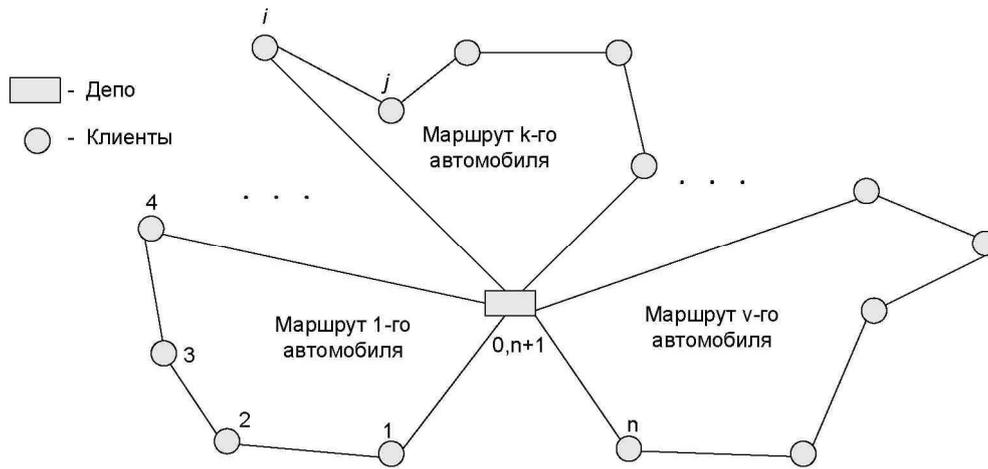


Рис. 1. Графическое представление ТЗ

Исходя из приведенных выше обозначений математическая формулировка ТЗ с ограничением по времени следующая, нужно минимизировать целевую функцию (1), при ограничениях (2) – (9).

$$\sum_{k \in V} \sum_{(i, j) \in A} c_{ij} X_{ij}^k \quad (1)$$

$$\sum_{k \in V} \sum_{j \in N} X_{ij}^k = 1, \forall i \in C \quad (2)$$

$$\sum_{i \in C} d_i \sum_{j \in N} X_{ij}^k \leq q, \forall k \in V \quad (3)$$

$$\sum_{j \in N} X_{0j}^k = 1, \forall k \in V \quad (4)$$

$$\sum_{j \in N} X_{ih}^k - \sum_{j \in N} X_{hj}^k = 0, \forall h \in C, \forall k \in V \quad (5)$$

$$\sum_{j \in N} X_{i, n+1}^k = 1, \forall k \in V \quad (6)$$

$$\sum_{j \in N} X_{ij}^k (S_i^k + t_{ij} - S_j^k) \leq 0, \forall (i, j) \in A, \forall k \in V \quad (7)$$

$$a_i \leq S_i^k \leq b_i, \forall i \in N, \forall k \in V \quad (8)$$

$$X_{ij}^k \in \{0, 1\}, \forall (i, j) \in A, \forall k \in V. \quad (9)$$

Целевая функция (1) определяет цену всех маршрутов всех транспортных средств (общая цена транспортного плана). Ограничение (2) полагает, что каждый клиент обслуживается только одним транспортным средством и только один раз. Ограничение (3) определяет, что транспортное средство не может обслужить больше клиентов, чем позволяет его грузоподъемность. Ограничение (4) означает, что каждый автомобиль покидает депо один раз. Ограничение (5) показывает, что автомобиль может покинуть вершину h , только если он прибыл в эту вершину. Аналогично ограничению (4), ограничение (6) означает, что все транспортные средства возвращаются в депо, причем один раз. Это ограничение следует из ограничений (4) и (5). Ограничение (7) означает, если автомобиль движется из вершины i в j , то время прибытия автомобиля в j не может быть меньше суммы времени прибытия автомобиля в пункт i (S_i^k) и времени движения автомобиля из пункта i в пункт j (t_{ij}). Ограничение (8) – это ограничение по времени, прибытие автомобиля к клиенту должно быть в пределах временного окна.

Как говорилось выше, первичная цель минимизация общего количества требуемого автотранспорта, вторичная, при одном и том же количестве транспортных средств минимизация общего количества пройденного пути или общего затраченного

времени (минимизация общей стоимости маршрутов) [3]. Временные ограничения делятся на “мягкие” (soft time windows) – нарушать которые разрешается (нарушение таких ограничений наказывается штрафом, выраженным в количественной форме и добавляемом к ЦФ) и “жесткие” (hard time windows) – нарушение которых не допускается ни при каких обстоятельствах [2]. Как правило, алгоритмы для “жестких” и “мягких” ограничений различаются.

3. Динамическая транспортная задача с ограничением по времени

В отличие от статической ТЗ с ограничением по времени, параметры которой известны до начала ее решения и не изменяются, ни во время ее решения, ни во время выполнения, в динамической ТЗ с ограничением по времени, условия задачи могут меняться во время двух перечисленных выше этапов. Возникает новый класс задач – динамические транспортные задачи с ограничением по времени (DVRPTW) [5]. Далее рассмотрим основные параметры, изменение которых превращает статическую ТЗ с ограничением по времени в динамическую. Согласно [4] и [5] в литературе выделяют два главных динамических элемента ТЗ это – *динамические запросы клиентов* и *динамическое время движения* от одного клиента к другому. *Динамические запросы клиентов* – это новые запросы, которые возникают во время решения задачи и которые должны быть выполнены. *Динамическое время движения* – это время движения от одного клиента к другому, которое может изменяться в пределах какого-либо интервала, либо изменяться заранее (до начала решения задачи) не известным образом. На изменение времени движения между двумя клиентами могут влиять различные причины: дорожный трафик, случайные аварии, погода, местоположение и конструкция дорог и т. д. Данные причины в различной степени влияют на возможность предсказания времени движения, как правило, различают три типа *динамического времени движения*:

Время движения можно предсказать заранее (long-term forecasts). Время движения заведомо зависит от времени суток или от дня недели (time-depend) [4]. Хотя данное время и изменяется, эти изменения известны заранее и могут быть описаны до начала решения задачи и, по существу, является статическим [5].

Время движения может быть предсказано на короткий промежуток времени (short-term forecasts), когда имеется более полная информация о текущей ситуации на дороге [5]. Считается что такую информацию, например, о времени движения автомобиля из пункта i в пункт j можно получить только по прибытии автомобиля в пункт i [4].

Время движения не может быть предсказано и рассматривается стохастическим [5]. Причиной изменения данного времени могут стать непредвиденные аварии и поломки в пути, а также прочие случайности на дорогах, это время является истинно динамическим компонентом времени передвижения.

Динамические элементы ТЗ, выделяемые в математической модели динамической транспортной задачи, являются её важной и неотъемлемой частью. При описании математической модели задачи определяется, какой вид динамизма рассматривается в данной задаче. Как правило, отдельный алгоритм решения динамической ТЗ с ограничением по времени принимает в рассмотрение один или несколько динамических элементов описанных выше [5].

4. Методы решения динамической транспортной задачи с ограничением по времени

Согласно литературе [1-3] методы решения ТЗ с ограничением по времени как статических, так и динамически можно разделить на *точные* (exact approaches), *эвристические* (heuristic approaches) и *метаэвристические* (metaheuristics). Далее рассмотрим методы, применяемые при решении статической версии ТЗ с ограничением по времени, придерживаясь именно этого разделения. Решение статической ТЗ с ограничением по

времени ключевой элемент решения динамической ТЗ с ограничением по времени, потому что динамическую задачу можно представить как серию статических задач, где очередная статическая задача возникает при изменении параметров текущей задачи [5]. Также нужно заметить, что будут рассмотрены лишь общие методы, применяемые в различных алгоритмах, а не сами алгоритмы. На основе данных методов в дальнейшем возможно синтезирование нового алгоритма решения динамической ТЗ с ограничением по времени.

ТЗ с ограничением по времени, как и ее динамический вариант, относится к классу NP-полных задач, точные методы решения такого класса задач основаны на полном переборе всех возможных решений, и являются не эффективными при решении задач большой размерности из-за их больших временных затрат (время решения экспоненциально зависит от размерности задачи). Поэтому, в данном обзоре они не будут рассматриваться, основное внимание уделим эвристическим и метаэвристическим методам. Недостаток данных методов в том, что они являются приближительными (approximate methods), решение, полученное данными методами, может быть далеко от оптимального. Преимущество в том, что они позволяют найти приемлемое решение NP-полных задач большой размерности за приемлемое время [3]. Далее рассмотрим два классических эвристических метода *метод конструирования маршрута* (route construction method) и *метод улучшения маршрута* (route improvement method), основанный на методе локального поиска.

4.1. Эвристические методы

Эвристические методы конструирования маршрута (route construction heuristics). Суть данных методов, в общем виде, представляется следующим образом. Построение маршрута осуществляется путем последовательного добавления еще не обслуженных клиентов к текущему маршруту (вставки клиентов в текущий маршрут). Для инициализации каждого нового маршрута выбирается вероятностным образом или по некоторому критерию (например, выбирается клиент наиболее удаленный от точки начала маршрута – депо) “первый” клиент (seed customer) из числа еще не обслуженных клиентов (un-routed customers) для включения его в текущий маршрут (partial route) [1]. Затем оставшиеся клиенты оцениваются по некоторым критериям, согласно этому критерию выбирается лучший из них и добавляется к текущему маршруту. Добавление клиента к маршруту не должно нарушать ограничений задачи (временных ограничений, ограничений на грузоподъемность автомобиля). Если нет допустимых клиентов (не обслуженных клиентов, которых можно добавить к текущему маршруту без нарушения ограничений задачи) начинается новый маршрут. Если все клиенты включены в маршруты, не обслуженных клиентов не осталось, алгоритм завершается.

Приведем наиболее известный метод конструирования маршрута предложенный Соломоном в 1987 году [1-3]. Инициализация маршрута начинается выбором первого клиента для включения в маршрут. Из числа еще не обслуженных клиентов выбирается, либо клиент наиболее удаленный от депо, либо клиент которого нужно обслужить раньше всех (согласно ограничениям по времени). Затем каждый допустимый не обслуженный клиент u оценивается на предмет вставки его между двух соседних клиентов текущего маршрута, по двум критериям $c_1(i, u, j)$ и $c_2(i, u, j)$, лучший клиент согласно этим критериям вставляется на маршрут между клиентами i и j . Первый критерий определяет лучшее место вставки в текущий маршрут, т. е. клиентов i и j между которыми будет вставлен клиент u и состоит из оценки добавочного расстояния $c_{11}(i, u, j)$ и времени $c_{12}(i, u, j)$ при добавлении клиента u между клиентами i и j . Для каждого клиента u критерий $c_1(i, u, j)$ считается следующим образом:

$$c_1(i, u, j) = a_1 c_{11}(i, u, j) + a_2 c_{12}(i, u, j), \quad (10)$$

где $a_1 + a_2 = 1$, $a_1 \geq 0$, $a_2 \geq 0$,

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - m d_{ij}, \quad m \geq 0, \quad (11)$$

$$c_{12}(i, u, j) = b_{iu} - b_j. \quad (12)$$

Где d_{iu} , d_{uj} , d_{ij} – расстояния между клиентами i и j , u и j , i и j соответственно.

Второй критерий определяется для каждого клиента u при условии, что он будет вставлен на маршрут между клиентами i и j , определенными на основе первого критерия, следующим образом:

$$c_2(i, u, j) = \lambda d_{ou} - c_1(i, u, j), \quad \lambda \geq 0. \quad (13)$$

Вычислительная сложность алгоритма Соломона согласно [1] $O(n^3)$. Данный алгоритм широко используется в последующих алгоритмах для их инициализации (например, для построения начального – опорного решения с целью его последующего улучшения – *метод улучшения маршрута*). В 1993 году был предложен метод *параллельного конструирования маршрута*. Данный метод основан на описанном выше методе, отличие параллельного метода от последовательного в том, что оценка вставки каждого клиента в маршрут происходит не последовательно, а параллельно (одновременно), следовательно, время выполнения алгоритма сокращается [3].

Данные алгоритмы являются быстрыми, но решение, получаемое с помощью этих алгоритмов, является, как правило, низкого качества в смысле целевой функции. И как говорилось выше, *методы конструирования маршрута* в основном используются как инициализационные для других более сложных методов, которые будут рассмотрены ниже.

Эвристические методы улучшения маршрута (route improvement methods). Данные методы относятся к классическим методам локального поиска. Суть методов состоит в следующем. Имеется текущее решение задачи (набор допустимых маршрутов). Данное решение итеративно сравнивается с модифицированным решением из окрестности текущего решения. Если модифицированное решение лучше текущего, в смысле целевой функции, то модифицированное решение становится текущим. Набор модифицированных решений, которые можно получить из текущего называется *окрестностью* текущего решения. Механизм получения модифицированного решения из текущего является важным элементом метода, в транспортных задачах для этого часто используется перестановка r -дуг между маршрутами или внутри маршрутов, следует заметить, что рассматриваются только допустимые решения. Алгоритм заканчивает свою работу, если r -оптимальное решение найдено. r -оптимальное решение – это решение, которое не может быть улучшено дальнейшими r -перестановками [2]. Проверка r -оптимальности требует времени порядка $O(n^r)$ [3]. Качество полученного решения зависит от *начального решения* и от используемого механизма получения *окрестности* данного решения [3]. Далее рассмотрим один из методов данного типа – метод перестановки дуг. Метод перестановки дуг, названный λ -перестановками (λ -interchange), был впервые предложен Османом в 1989 году [6]. Данный метод основан на перестановке клиентов между маршрутами. Перестановка клиентов может осуществляться как перемещением клиентов с одного маршрута на другой, так и перестановкой (взаимным обменом) клиентов между двух маршрутов. Для перестановки клиентов применяются операторы следующего вида: оператор $(I, 0)$ для маршрутов (R_p, R_q) означает перемещение одного клиента с маршрута q на маршрут p , оператор $(1, 1)$ обмен двух клиентов между маршрутами q и p . Таким образом, работа алгоритма следующая. Для заданной задачи генерируется набор допустимый маршрутов методом Соломона, описанным выше. Новое решение из окрестности текущего получаем при помощи операторов перестановки. Если новое решение лучше текущего, то текущее решение заменяется новым. Процесс продолжается до тех пор, пока не будет исследована вся окрестность решения [6].

В последующее годы было предложено множество методов основанных на перестановках клиентов между маршрутами [3]. Данные методы дают более качественное решение, чем методы конструирования маршрута, но являются более затратными по времени.

4.2. Метаэвристические методы

В последние годы для решения ТЗ с ограничением по времени широко используются так называемые метаэвристические методы. К данным методам относятся: метод “модельной” закалки или метод моделирования отжига (simulated annealing), метод поиска с запретами (tabu search), метод муравьиной колонии (ant colony optimization), генетические и эволюционные алгоритмы [2]. Преимущество данных методов перед классическими эвристическими методами, основанными на методе локального поиска, в том, что они позволяют исследовать большее пространство поиска для нахождения решения близкого к оптимальному, тогда как методы локального поиска останавливаются после нахождения локального решения выбранной окрестности. Рассмотрим далее один из алгоритмов поиска с запретами, т. к. данный метод широко используется при решении ТЗ с ограничением по времени [7]. Привлекательность данного метода в том что, хотя в его основе и лежит метод локального поиска, метод запретов позволяет продолжать поиск после нахождения локального оптимума, тем самым, расширяя пространство поиска, в надежде найти решение близкое к оптимальному. Для этой цели метод запретов использует *список запретов* (tabu list), именно этот основной элемент метода запретов используется для предотвращения заикливания (возвращение к предыдущим решениям) и для расширения пространства поиска [2]. Другими основными понятиями в методе поиска с запретами, заимствованными из методов локального поиска, являются *пространство поиска* (search space) и *механизм построения окрестности решения* (neighborhood structure). Рассмотрим эти понятия более подробно. *Пространство поиска* – это все возможные решения, которые могут быть получены в ходе решения задачи, в данном случае маршруты транспортных средств. Пространство поиска тесно связано с механизмом построения окрестности данного решения. Окрестностью текущего решения называют все решения, которые могут быть получены из текущего, путем применения к текущему решению определенных операторов, процесс применения этих операторов и называется построением окрестности текущего решения [9]. *Список запретов* – это ключевое понятие, отличающее метод запретов от методов локального поиска. В списке запретов содержатся перемещения (moves) которые применялись на предыдущих этапах решения. Список запрещенных перемещений может быть как фиксированной, так и переменной длины. Время, в течение которого данное перемещение содержится в списке запретов, называется продолжительностью запрета (tabu tenure). По истечении этого времени перемещение удаляется из списка запретов. Однако список запретов может оказать слишком жесткое ограничение на процесс поиска, он может запрещать перемещения, которые не приведут к заикливанию алгоритма, а наоборот являются предпочтительными. В данном случае используется способ, который называется аспирационный критерий (aspiration criteria). Суть метода заключается в том что, перемещение допускается, даже если оно находится в списке запретов, если решение, полученное при применении данного перемещения, лучше в смысле ЦФ текущего оптимального решения, соответственно полученное решение становится текущим оптимальным решением. Описанные выше элементы являются основными в методе поиска с запретами [9]. В работе [11] предлагается алгоритм решения динамической ТЗ с ограничением по времени с использованием параллельного метода поиска с запретами. Преимущества параллельного метода состоит в том, что при распараллеливании вычислений уменьшается время работы алгоритма. Рассмотрим данный алгоритм более подробно. Алгоритм состоит из следующих этапов.

Начало

1. Методом конструирования маршрута Соломона строится p различных решений задачи. Для выбора начального клиента используется вероятностный метод.

2. К каждому из полученных p решений применяется метод поиска с запретами. Маршруты полученных в результате решений сохраняются в адаптивной памяти.

Пока $stop \neq true$ выполняем следующее

3. Из маршрутов содержащихся в адаптивной памяти строится новое решение, это решение называется текущим.

I итераций делаем следующее

4. Разделяем текущее решение на S различных подмножества маршрутов.

5. Применяем метод поиска с запретами к каждому подмножеству.

6. Строим полное решение соединением подмножеств маршрутов, полученное решение назовем текущим.

7. Сохраняем маршруты текущего решения в адаптивной памяти.

8. К каждому маршруту лучшего решения применяется процедура дополнительной оптимизации.

Конец

Рассмотрим каждый из пунктов подробно. В пункте 1 используется метод конструирования Соломона, рассмотренный выше. Выбор начальных клиентов для инициализации маршрутов происходит вероятностным образом [9]. Адаптивная память, используемая в данном алгоритме, представляет собой массив памяти размера M в котором хранятся маршруты лучших решений, полученных в результате работы алгоритма. Адаптивная память предназначена для конструирования новых начальных решений. Для этого из адаптивной памяти выбираются маршруты, которые и будут представлять собой новое начальное решение. Рассмотрим две процедуры связанные с адаптивной памятью это – помещение маршрутов в память и выборка маршрутов из памяти. Заполнение адаптивной памяти происходит следующим образом, если адаптивная память заполнена не полностью, то маршруты всех решения получаемых при работе алгоритма помещаются в адаптивную память, если память заполнена, то новое решение помещается в память только в том случае, если его ЦФ “лучше” ЦФ “худшего” решения в адаптивной памяти. Решения, в виде маршрутов, занимают позиции соответствующие их целевой функции (маршруты лучшего решения, в смысле ЦФ занимают первые позиции). Выборка маршрутов из адаптивной памяти для конструирования нового решения происходит следующим образом. Вероятностным образом из памяти выбираются маршруты, вероятность выбора маршрута из памяти тем выше, чем меньше его порядковый номер в адаптивной памяти. При выборе очередного маршрута из памяти исключаются маршруты, включающие в себя клиентов из выбранного маршрута. Процедура выборки маршрутов продолжается до тех пор, пока не будет построено новое решение. Если адаптивная память становится пустой до окончания построения нового решения, то к оставшиеся не обслуженным клиентам применяется процедура конструирования маршрута Соломона, тем самым завершается процесс построения нового решения [9].

Процедура поиска с запретами применяется как к “целому” решению из адаптивной памяти на инициализационном этапе, так и к подмножеству маршрутов (части решения) полученному в результате разделения решения на S подмножеств. Процедура разделения решения на части и последующее восстановление целого решения путем объединения разделенных частей (decomposition/reconstruction) производится для уменьшения времени вычислений и ускорения процесса поиска. Сама процедура поиска с запретами выглядит следующим образом [9]:

Пусть имеется некоторое начальное решения (набор маршрутов) покрывающий некоторое количество клиентов, назовем его текущим.

Начало

1. Итераций делаем следующее

2. Выбираем лучшее решение из окрестности текущего решения.
3. Если данное решение не запрещено (данного решения нет в списке запретов) оно становится текущим.
4. Если данное решение лучше, в смысле ЦФ, “лучшего” решения найденного на протяжении всей работы алгоритма, то клиенты внутри каждого маршрута текущего решения переупорядочиваются с использованием метода Соломона и список запретов обновляется.
5. Возвращается “лучшее” решение.

Конец

Для повышения быстродействия параллельно запускаются несколько дочерних потоков, которые управляются главным потоком. Главный поток занимается диспетчеризацией дочерних потоков. Этот поток извлекает решения из адаптивной памяти и передает решения дочерним потокам. Дочерние потоки после обработки решения методом поиска с запретами возвращают его главному потоку, который принимает решение о помещении решения в адаптивную память [10]. Для повышения качества решения и уменьшения времени работы алгоритма используется разделение на подзадачи, с последующим восстановлением решения, и внутри маршрутная оптимизация [9-10].

В данном методе динамическая ТЗ с ограничением по времени моделируется как серия статических задач. Добавление нового заказа приводит к появлению нового не обслуженного клиента, данный клиент вставляется во все решения, которые находятся в адаптивной памяти и алгоритм продолжает свою работу с новым количеством клиентов [11]. Данный алгоритм входит в число лучших алгоритмов для решения ТЗ с ограничением по времени большой размерности [7].

Заключение

Оценка алгоритмов как эвристических, так и метаэвристических для NP-полных задач, для ТЗ с ограничением по времени не является тривиальной задачей. Важным свойством всех эвристических алгоритмов является время выполнения и близость полученного решения к оптимальному. Обычно в данных алгоритмах идет размен между качеством получаемого решения и временем выполнения алгоритма. Чем большее количество итераций проходит алгоритм, тем более качественное решение мы получаем. Наиболее общий метод сравнения алгоритмов эмпирический. Но при сравнении алгоритмов опытным путем исследователи сталкиваются с различными используемыми вычислительными ресурсами и типами решаемых задач. Так же на время выполнения влияет разный профессиональный уровень кодирования алгоритма. Некоторые алгоритмы, показав себя хорошо на одних типах задач, не справляются с другими. Для задач типа ТЗ с ограничением по времени широко используются методы сравнения на основе решения 56-ти тестовых задач (benchmark problems) предложенных Соломоном в 1987 году [3, 8]. Эти задачи имеют сто клиентов, одно центральное депо, ограничение по грузоподъемности, ограничения по времени. Данные задачи разделяются на классы. В задачах класса C1 и C2 клиенты располагаются группами (кластерами), в задачах класса R1 и R2 клиенты располагаются случайным образом и соответственно в задачах класса RC1 и RC2 клиенты располагаются смешанно как группами, так и случайным образом. Имеются задачи с 25%-ым, 50%-ым, 75%-ым и 100%-ым количеством клиентов с ограничениями по времени. Тестирование алгоритмов для решения ТЗ с ограничением по времени на данных тестовых задачах показало, что качество решений полученных с использованием метаэвристических алгоритмов превосходит качество решений полученных с помощью

классических конструктивных методов и методов локального поиска. Метаэвристические методы более затратные по времени и более сложные в применении, однако именно они являются наиболее эффективными в решении практических задач [8]. Следует отметить, что метаэвристические методы не противопоставляются классическим эвристическим методам, а наоборот используют их для формирования стратегии поиска в рамках конкретного метаэвристического алгоритма. Метаэвристические методы являются дальнейшим развитием эвристических методов, и в связи с полученными результатами являются перспективным направлением в решении оптимизационных задач.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Tobias Babb. Pickup and Delivery Problem with Time Windows, Coordinated Transportation Systems: The State of the Art. Department of Computer Science University of Central Florida Orlando, Florida, 2005.
2. J.-F. Cordeau, Guy Desaulniers, Jacques Gesrosiers, Marius M. Solomon, Francois Soumis. The VRP with Time Windows. Chapter 7, Paolo Toth and Daniel Vigo (eds), SIAM, Monographs on Discrete Mathematics and Applications, 2001.
3. Olli Bräysy, Michel Gendreau. Route Construction and Local Search Algorithms for the Vehicle Routing Problem with Time Windows. Internal Report STF42 A01024, SINTEF Applied Mathematics, 2001.
4. Jean-Yves Potvin, Ying Xu, Ilham Benyahia. Vehicle routing and scheduling with dynamic travel times. Computer & Operations Research 33, p. 1129 – 1137, 2006.
5. Truls Flatberg, Geir Halse, Oddvar Kloster, Eivind J. Nilssen, Atle Riise. Dynamic and Stochastic Aspects in Vehicle Routing – A Literature Survey. STF90 A05413, SINTEF Applied Mathematics, 2005.
6. K. C. Tan, L. H. Lee, K. Q. Zhu. Heuristic Methods for Vehicle Routing Problem with Time Windows. Proceedings of the 6th International Symposium on Artificial Intelligence & Mathematics, Ft. Lauderdale, Florida, 2000.
7. Olli Bräysy, Michel Gendreau. Tabu Search Heuristics for the Vehicle Routing Problem with Time Windows. Internal Report STF42 A01022, SINTEF Applied Mathematics, 2001.
8. Michel Gendreau, Olli Bräysy. Metaheuristic Approaches for the Vehicle Routing Problem with Time Windows: A Survey. MIC2003: The Fifth Metaheuristics International Conference, Kyoto, Japan, August 25-28, 2003.
9. Eric Taillard, Philippe Badeau, Michel Gendreau, Francois Guertin and Jean-Yves Potvin. A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. Transportation Science 31, 170 – 186, 1997.
10. Eric Taillard, Philippe Badeau, Michel Gendreau, Francois Guertin and Jean-Yves Potvin. A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows. Transportation Research – C 5, 109 – 122, 1997.
11. Gendreau M., et al., Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. Transportation Science, 33 (4), 381 – 390, 1999.