УДК 656.13:004.023

А.И. Маслеев, А.Д. Кулязин, А.В. Липенков

ТЕСТИРОВАНИЕ ЭВРИСТИКИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ ТРАНСПОРТА С РАЗДЕЛЬНОЙ ДОСТАВКОЙ

Нижегородский государственный технический университет им. Р.Е. Алексеева

Рассматривается один из эвристических методов решения задачи маршрутизации транспорта с раздельной доставкой (SDVRP). Представлен алгоритм метода, его математическая модель и параметры. Приведены этапы алгоритмизации в профессиональной среде моделирования AnyLogic. Порядок объезда пунктов внутри получаемых маршрутов оптимизировался методом «ветвей и границ».

Ключевые слова: задача маршрутизации транспорта, vehicle routing problem, метод раздельной доставки, метод «ветвей и границ», Anylogic.

Рынок грузовых перевозок в современном мире представлен огромным количеством транспортных компаний. Один из путей максимизации их прибыли – сокращение издержек в цепи поставок, в частности, путем составления наиболее рациональных маршрутов доставки грузов. Задача составления таких маршрутов известна как проблема маршрутизации транспортных средств (*Vehicle Routing Problem, VRP*) [1]. Ее исследования представлены в работах А.О. Алексеева, М.Ю. Ахлебинского, И.И. Меламеда, С.И. Сергеева, И.Х. Сигала, *G. Clarke, D. Wright, M. Dror, M. Gendreau, A. Hertz, G. Laporte, I.H. Osman, J.L. Blanton, G. Jeon, B. Bullnheimer* и др.

Выделяют различные подклассы задачи маршрутизации [2]:

- *Capacitated VRP (CVRP)* – каждое транспортное средство имеет ограниченную грузоподъемность [3];
- *VRP with Time Windows (VRPTW)* – каждый клиент должен быть обслужен в определенное временное окно [4];
- *Multiple Deport VRP (MDVRP)* – используются несколько депо для обслуживания клиентов [5];
- *VRP with Pick–Ups and Delivering (VRPPD)* – клиенты могут возвращать некоторые товары в депо [6];
- *VRP with Satellite Facilities* – существует возможность дозагрузки транспортного средства на маршруте [7].

Одной из интересных разновидностей *VRP* служит задача *SDVRP,* формулируемая следующим образом. Задана «транспортная» сеть, состоящая из узлов – мест расположения клиентов и спрос каждого из них. Выезд всех транспортных средств осуществляется из депо, как правило имеющего нулевой индекс. Необходимо составить кольцевые маршруты (с началом и окончанием в депо) таким образом, чтобы полностью удовлетворить спрос всех клиентов, при этом минимизировав общее пройденное расстояние. Главным ее допущением по сравнению с классической *VRP* является возможность посещения одного клиента неограниченное количество раз (*split delivery*). Впервые официально, задача *SDVRP* была представлена *Dror M.* и *Trudeau P.* в 1990 г. [8]. В данной работе отмечалось, что в случае, когда имеются клиенты с потребностью в 10 % и более от грузоподъемности автомобиля, разделение доставок начинает существенно экономить пробег транспортных средств.

В настоящее время проблеме *SDVRP* посвящено большое количество исследований [9]. В одном из них *J.H. Wilck* и *T.M. Cavalier* [10] предложили эвристику – «конструктор», состоящий из трех основных управляющих процедур, в итоге приводящих к 72 вариантам решения.

Целью настоящей работы является алгоритмизация рассматриваемой эвристики и проверка качества получаемых по ней решений. Алгоритм был разработан на языке программирования *Java* в профессиональной среде имитационного моделирования *Anylogic*.

Рассмотрим эвристику более подробно. Ниже представлены исходные данные, переменные и описание эвристики.

Наборы индексов:

$i = \{1, 2, \cdots, n\}$ – индекс клиента (узла);

$j = \{1, 2, \cdots, n\}$ – индекс клиента (узла);

$k = \{1, 2, \cdots, m\}$ – индекс маршрута.

Входные параметры:

m – количество маршрутов транспортных средств, необходимое для удовлетворения спроса всех клиентов;

n – количество клиентов (узлов);

Q – грузоподъемность единицы подвижного состава;

c_{ij} – расстояние от клиента i до клиента j;

d_i – требование клиента i (где $d_1 = 0$, поскольку потребность депо равна 0);

Переменные:

y_{ik} – переменная-массив, отвечающая за посещение клиента i по маршруту k;

v_{ik} – переменная-массив, обозначающая количество груза, доставленного клиенту i по маршруту k;

S – общая доступная «запасная мощность» ($S = mQ - D; D = \sum_{i=1}^{n} d_i$);

A – средняя «запасная мощность» на маршрут (т.е., $A = \frac{S}{m}$, округленная до ближайшего целого);

R – набор клиентов с неудовлетворенным спросом;

P_k – набор клиентов, входящих в маршрут k;

f_i – количество неудовлетворенных потребностей, оставшихся для узла i;

b_k – остаточная грузоподъемность транспортного средства k.

При решении необходимо минимизировать функцию Z:

$$Minimize\ Z = \sum\nolimits_{i=1}^{n} \sum\nolimits_{\substack{j=1 \\ i \neq 1}}^{n} c_{ij} \sum\nolimits_{k=1}^{m} x_{ijk} \quad (1)$$

Также на модель накладываются следующие ограничения:

$$\sum\nolimits_{k=1}^{m} v_{ik} < d_i,\ \forall\ i = 2, \ldots, n \quad (2)$$

$$\sum\nolimits_{i=2}^{n} < Q,\ \forall\ k = 1, \ldots, m \quad (3)$$

$$\sum\nolimits_{\substack{i=1 \\ i \neq p}}^{n} x_{ijk} - \sum\nolimits_{\substack{j=1 \\ j \neq p}}^{n} x_{ijk} = 0,\ \forall k = 1,\ldots,m;\ p = 1,\ldots,n \quad (4)$$

$$u_{ik} - u_{jk} + nx_{ijk} \leq n - 1,\ \forall i = 2,\ldots,n;\ i \neq j;\ k = 1,\ldots,m \quad (5)$$

$$d_i y_{ik} \geq v_{ik},\ \forall k = 1,\ldots,m;\ i = 2,\ldots,n \quad (6)$$

$$\sum\nolimits_{\substack{j=1 \\ j \neq i}}^{n} x_{ijk} = y_{ik},\ \forall k = 1,\ldots,m;\ i = 2,\ldots,n \quad (7)$$

$$\sum\nolimits_{j=2}^{n} (x_{1ij} + x_{i1k}) = 2,\ \forall k = 1,\ldots,m \quad (8)$$

$$x_{ijk} \in \{0,1\},\ \forall i = 1,\ldots,n;\ j = 1,\ldots,n;\ i \neq j;\ k = 1,\ldots,m \quad (9)$$

$$y_{ik} \in \{0,1\},\ i = 2,\ldots,n;\ k = 1,\ldots,m \quad (10)$$

$$v_{ik} \geq 0;\ \forall i = 2,\ldots,n;\ k = 1,\ldots,m \quad (11)$$

Цель задачи (1) заключается в минимизации общего пройденного расстояния. Ограничения (2) и (3) гарантируют, что весть потребительский спрос будет удовлетворен без нарушения вместимости транспортного средства. Ограничения (6) и (7) заставляют переменные быть положительными, если груз доставлен клиенту *i* по маршруту *k*. Ограничение (8) гарантирует, что каждый маршрут будет начинаться и заканчиваться в депо.

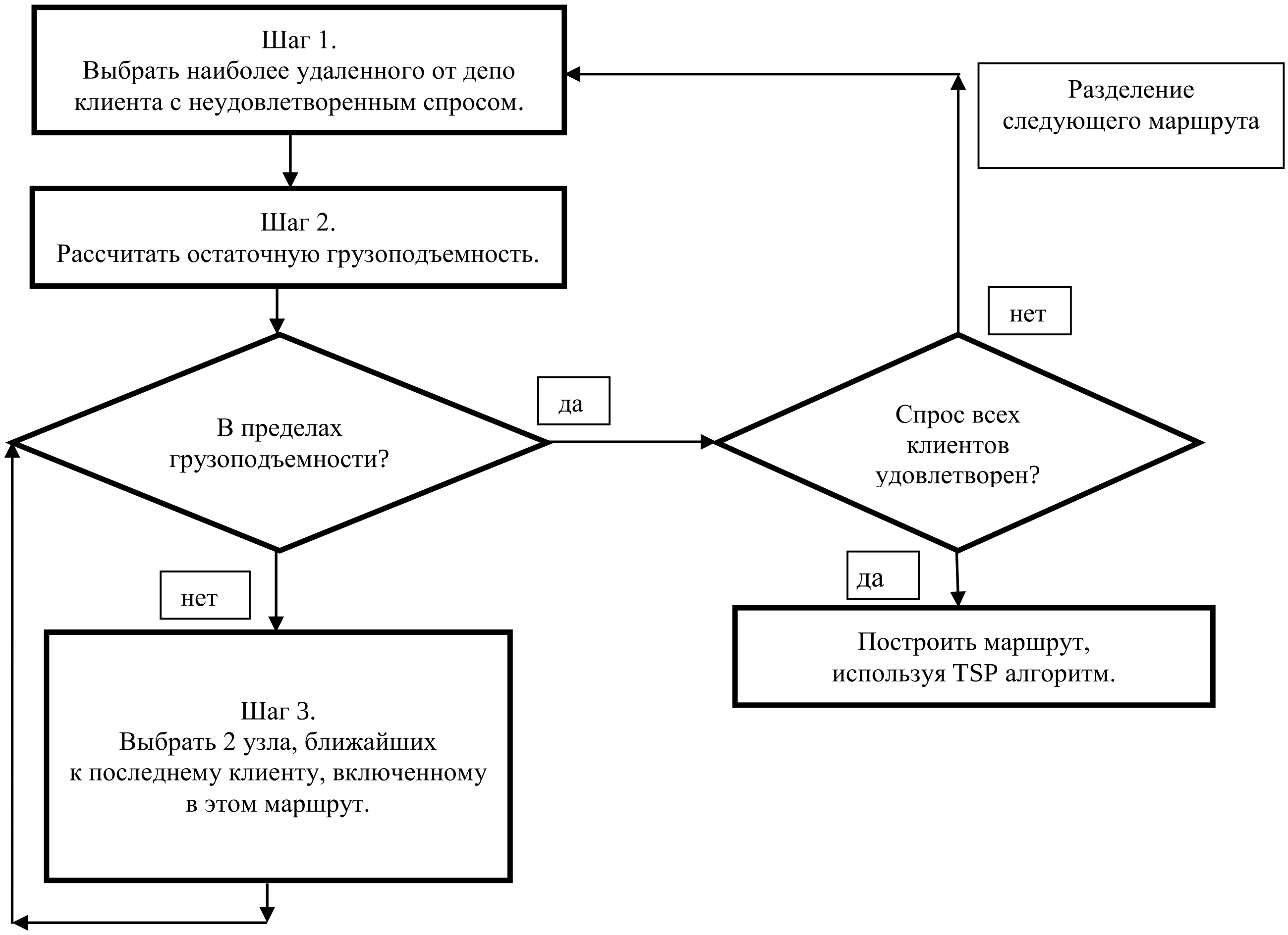

Рис. 1. Блок-схема эвристики для решения задачи SDVRP

Блок-схема алгоритма представлена на рис. 1.

Шаг 0: инициализация данных и переменных.

На данном шаге определяются константы *m, n, D, Q, K* и *S*, где *D* – суммарный спрос всех клиентов, *K* – теоретически необходимое число транспортных средств.

Вычисляется начальное значение *A*. Если *m* <*K*, где $K = \frac{D}{Q}$, то решение невозможно.

Все клиенты помещаются в набор *R*, кроме узла 1, который соответствует депо. Необходимо убедится, что P_k пуст для всех *k*. Полагается $f_i = d_i$ для всех *i*. Задается $k = 1$.

Шаг 1: выбор клиента для добавления в маршрут. Данный выбор идет из числа клиентов с неудовлетворенным спросом. Этот начальный клиент выбирается с использованием одного из следующих правил: самый удаленный клиент от депо, ближайший клиент к депо, срединный клиент по отношению к расстоянию от депо, клиент с наибольшим спросом, клиент с наименьшим спросом, самый дальний клиент на самом коротком расстоянии от соседнего клиента (т.е., перегон (*i, j*), где $i \neq j$, $i \neq 1$, $j \neq 1$, для которых l_{ij} – наименьшее), случайный клиент. В данной работе использован вариант с нахождением самого удаленного от депо клиента.

Если $k = m+1$, то *STOP*. Из множества *R* находится узел, наиболее удаленный от депо, ему присваивается индекс *i*. Узел *i* помещается в набор P_k. Если $f_i > Q$, тогда $y_{ik} = 1$, $v_{ik} = Q$, $f_{ik} = f_i - Q$, $A = \frac{S}{m-k}$ и $k = k+1$. Отправляемся к шагу 1. В противном случае, если $Q - f_i \le A$, удаляется *i* из множества *R* и устанавливается $y_{ik} = 1$, $v_{ik} = f_i$, $S = S - (Q - f_i)$, $A = \frac{S}{m-k}$, $f_i = 0$ и $k = k+1$ и отправляемся к шагу 1. В противном случае, удаляется *i* из множества *R* и устанавливается $y_{ik} = 1$, $v_{ik} = f_i$, $b_k = Q - f_i$ и $f_i = 0$, отправляемся к шагу 2.

Шаг 2: эвристика [10] предполагает возможность использования «порога мощности», что позволяет использовать «запасную мощность» транспортного средства (разницу между суммарной грузоподъемностью всех автомобилей и суммарный спрос всех клиентов, т.е., $mQ - D$). Этот порог рассчитывается по трем различным правилам. Эти правила могут быть проигнорированы, если запасная способность транспортного средства отсутствует, поскольку в этом случае все транспортные средства должны быть заполнены до максимальной емкости. В данной работе при разработке алгоритма использовался именно этот вариант.

На втором шаге выбираются узлы для добавления к формируемому маршруту.

Если множество *R* пусто, то *STOP*. Если в наборе *R* имеется только один узел, ему присваивается индекс *i* и осуществляется переход к шагу *2f*. В противном случае из множества *R* находится два ближайших клиента к последнему клиенту в множестве P_k. Первый ближайший клиент обозначается – *i*, второй – *j*.

Если $f_i > b_k$ и $f_j \le b_k$, осуществляется переход к шагу *2a*;

Если $f_i \le b_k$ и $f_j > b_k$, осуществляется переход к шагу *2b*;

Если $f_i \le b_k$; $f_j \le b_k$ и $f_i + f_j > b_k$, осуществляется переход к шагу *2c*;

Если $f_i \le b_k$; $f_j \le b_k$ и $f_i + f_j \le b_k$, осуществляется переход к шагу *2d*;

Если $f_i > b_k$ и $f_j > b_k$, осуществляется переход к шагу *2e*;

Шаг 2a: добавляется узел *j* к маршруту, который в настоящее время строится, а затем определяется необходимо ли учитывать другие узлы.

Если $b_k - f_j \le A$, необходимо удалить *j* из множества *R*, переместить *j* в P_k и установить $y_{jk} = 1$, $v_{jk} = f_j$, $S = S - (b_k - f_j)$, $A = \frac{S}{m-k}$, $f_j = 0$ и $k = k+1$ и отправится к шагу 1. В противном случае удалить *j* из множества *R*, переместить *j* в P_k и установить $y_{jk} = 1$, $v_{jk} = f_j$, $b_k = b_k - f_j$ и $f_j = 0$, и отправится к шагу 2.

Шаг 2b: добавляется узел *i* к маршруту, который в настоящее время строится, а затем определяет необходимо ли учитывать другие узлы.

Если $b_k - f_i \le A$, тогда необходимо удалить *i* из множества *R*, переместить *i* в P_k и установить $y_{ik} = 1$, $v_{ik} = f_i$, $S = S - (b_k - f_i)$, $A = \frac{S}{m-k}$, $f_i = 0$, $k = k+1$ и отправится к шагу 1. В противном случае удалить *i* из множества *R*, переместить *i* в P_k и установить $y_{ik} = 1$, $v_{ik} = f_i$, $b_k = b_k - f_i$, и $f_i = 0$, и отправится к шагу 2.

Шаг 2c: добавляется узел *i* к маршруту, который в настоящее время строится, а затем определяет необходимо ли учитывать другие узлы.

Если $f_i = b_k$, тогда необходимо удалить *i* из множества *R*, переместить *i* в P_k и установить $y_{ik} = 1$, $v_{ik} = f_i$, $A = \frac{S}{m-k}$, $f_i = 0$, $k = k+1$ и отправится к шагу 1. В противном случае удалить i из множества R, установить i в P_k , $y_{ik}=1$, $y_{jk}=1$, $v_{ik}=b_k-f_i$, $A=\frac{S}{m-k}$, $f_i=0$, $f_j=f_j-v_{jk}$, $k=k+1$ и отправится к шагу 1.

Шаг 2d: добавляются узлы i и j к маршруту, который в настоящее время строится, а затем определяет необходимо ли учитывать другие узлы.

Если $b_k-f_i-f_j\le A$, i и j удаляются из множества R, помещаются в P_k , и устанавливается $y_{ik}=1$, $y_{jk}=1$, $v_{ik}=f_i$, $v_{jk}=f_j$, $S=S-(b_k-f_i-f_j)$, $A=\frac{S}{m-k}$, $f_i=0$, $f_j=0$, $k=k+1$ и отправляемся к шагу 1. В противном случае удаляются i и j из множества R, помещаются в P_k и устанавливается: $y_{ik}=1$, $y_{jk}=1$, $v_{ik}=f_i$, $b_k=b_k-f_i-f_j$, $f_i=0$ и $f_j=0$, отправляемся к шагу 2.

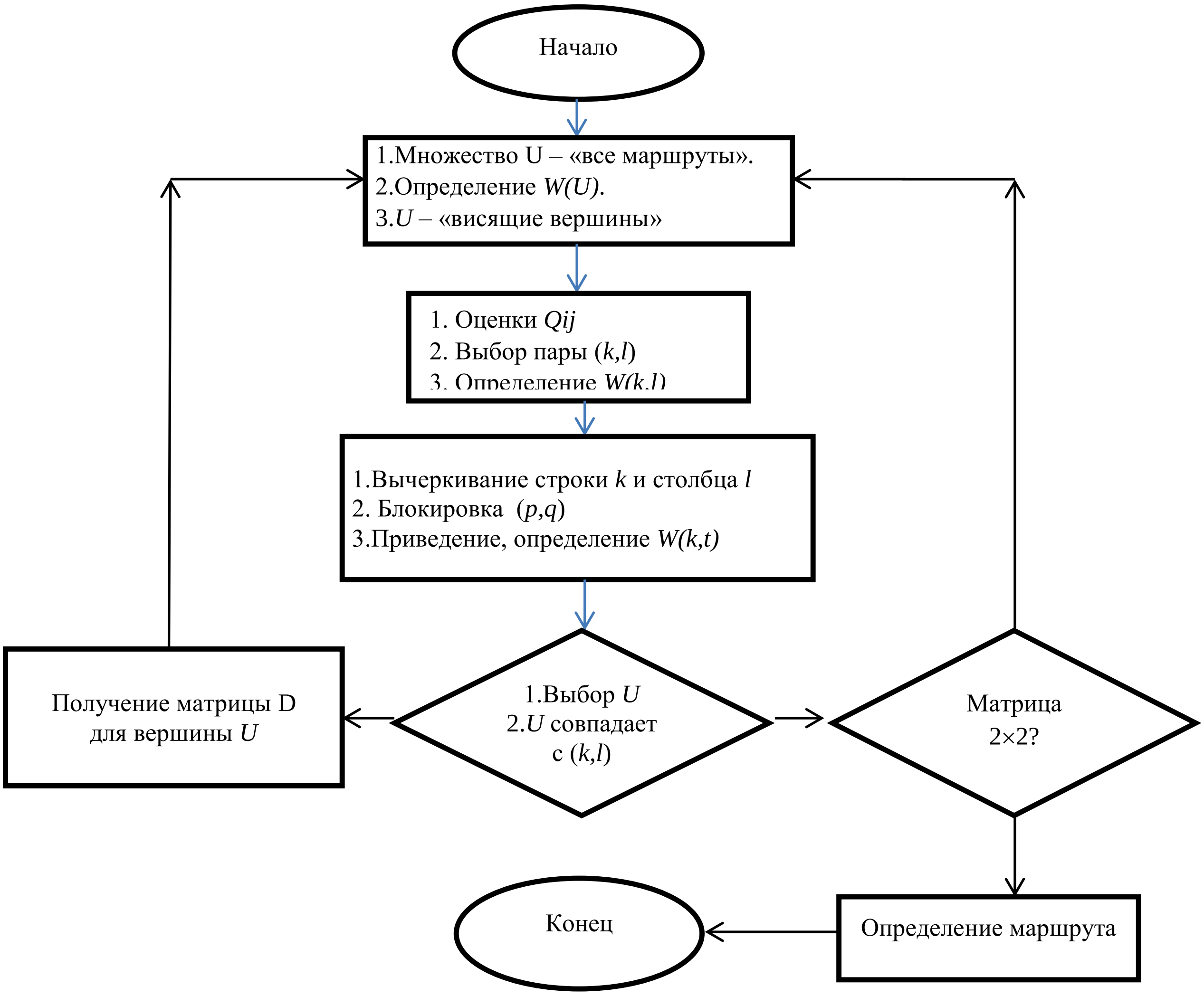

Рис. 2. Блок-схема метода «ветвей и границ»

Шаг 2е: добавляется узел i к маршруту, который в настоящее время строится, а затем определяет необходимо ли учитывать другие узлы.

Помещается i в P_k и устанавливается $y_{ik} = 1$, $v_{ik} = f_i - b_k$, $A = \dfrac{S}{m-k}$, $k = k+1$ и отправляемся к шагу 1.

Шаг 2f: добавляется узел i к маршруту, который в настоящее время строится, а затем определяет необходимо ли учитывать другие узлы.

Помещается i в P_k. Если $f_i \le b_k$, тогда необходимо удалить i из множества R и установить $y_{ik} = 1$, $v_{ik} = f_i$ и *STOP*. В противном случае назначить $y_{ik} = 1$, $v_{ik} = b_k$, $f_i = f_i - b_k$, $A = \dfrac{S}{m-k}$, $k = k+1$ и отправиться к шагу 1.

Рассмотренная эвристика не гарантирует оптимального порядка объезда пунктов внутри формируемых маршрутов, поэтому для каждого из них необходимо решать задачу коммивояжера. Данная задача решалась известным точным методом “ветвей и границ”, блок-схема которого представлена на рис. 2.

Пример визуализации решения в среде *Anylogic* представлен на рис. 3.

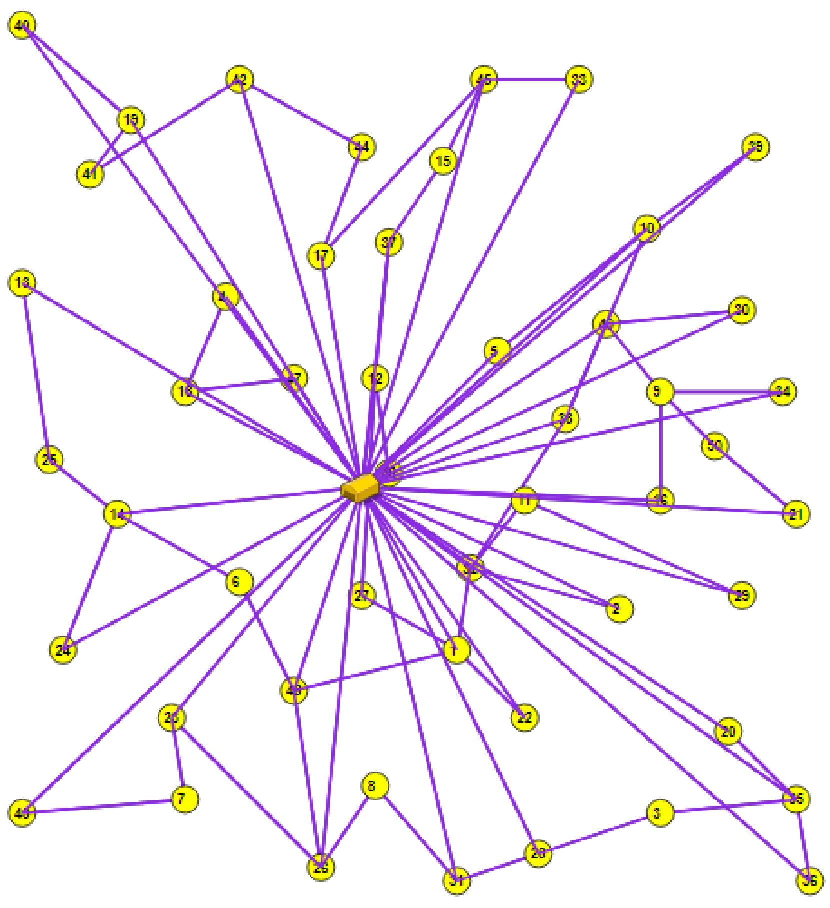

Рис. 3. Визуализиция решения

Тестирование эвристики проводилось на базе существующей библиотеке примеров, разработанных *Belenguer* [11] и *Chen* [12], для которых известны лучшие решения. Результаты моделирования представлены в табл. 1. Их анализ позволяет сделать вывод о том, что рассмотренная эвристика применима для решения задачи *SDVRP* и в ряде случаев дает решение близкое к оптимальному, однако в тоже время для некоторых примеров расхождения очень существенные и достигают 50 % и более по сравнению с лучшим решением.

Продолжение исследований видится в дальнейшем изучении эвристики, анализе ее вариаций и выработке рекомендаций по повышению качества получаемых по ней решений.

Таблица 1.
Результаты моделирования

Пример	Лучший результат	Полученный результат	Отклонение, %
eil22	375	492	-31,20
eil23	569	588	-3,34
eil30	510	627	-22,94
eil33	835	858	-2,75
eil51	521	618	-18,61
eilA76	832	983	-18,15
eilA101	817	1101	-34,76
eilB76	1023	1140	-11,44
eilB101	1077	1255	-16,52
eilC76	735	893	-21,50
eilD76	683	838	-22,69
S51D1	458	573	-25,10
S51D2	726	801	-10,33
S51D3	972	995	-2,36
S51D4	1677	1705	-1,66
S51D5	1440	1451	-0,76
S51D6	2327	2366	-1,67
S76D1	594	745	-25,42
S76D2	1147	1213	-5,75
S76D3	1474	1550	-5,15
S76D4	2257	2282	-1,1
S101D1	716	971	-35,61
S101D2	1393	1520	-9,11
S101D3	1975	2006	-1,56
S101D5	2915	2951	-1,23

Библиографический список

1. P. Toth and D. Vigo. The Vehicle Routing Problem. [Текст] // Society for Industrial and Applied Mathematics, Philadelphia, 2002.
2. G. B. Dantzig, J. H. Ramser. The Truck Dispatching Problem [Текст] // Management Science, Vol. 6, No. 1, 1959, pp. 80-91.
3. Augerat P., Belenguer J.M., Benavent E., Corber´an A., Naddef D. Separating Capacity Constraints in the CVRP Using Tabu Search [Текст]. European Journal of Oper. Res. – 1998. – 106. – P. 546-557.
4. Cordeau J.F., Laporte G., Mercier A. A unified tabu search heuristic for vehicle routing problems with time windows [Текст] Journal of the Oper. Res. Society. – 2001. – 52(8). – P. 928-936.
5. Min H., Current J., Schilling D. The Multiple Depot Vehicle Routing Problem With Backhauling [Текст] Journal of Business Logistics. – 1992. – 13 (1). – P. 259-288.
6. Chen J. F., Wu, T. H. Vehicle routing problem with simultaneous deliveries and pickups [Текст] // Journal of the Oper. Res. Society. – 2006. – 57. – P. 579-587.
7. Janathan F. Bard, Lui Huang, Moshe Dror, Patrick Jaillet. Branch and cut algorithm for the VRP with satellite facilities [Текст]. Rec. Feb.1997 and acc. Oct.1997.
8. M. Dror and P. Trudeau. Split Delivery Routing [Текст] Naval Research Logistics, Vol. 37, 1990, pp. 383-402.

9. Archetti C., Speranza M.G. An overview on the split delivery vehicle routing problem [Текст] // Operations research proceedings 2006. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operations Research (SVOR) Karlsruhe, September 6–8. – 2006 – pp. 123-127
10. Wilck J. H. IV, Cavalier T.M. A Construction Heuristic for the Split Delivery Vehicle Routing Problem [Текст]. – American Journal of Operations Research. – 2012. – V. 2. – pp. 153-162
11. J. M. Belenguer, M. C. Martinez, E. A. Mota. Lower Bound for the Split Delivery VRP [Текст]. Operations Research. – V. 48. – No. 5. – 2000. –pp. 801-810.
12. S. Chen, B. Golden, E. Wasil. The Split Delivery Vehicle Routing Problem: Applications, Test Problems and Computational Results [Текст]. Networks. – V. 94. – No. 4. – 2007. – pp. 318-329.