Common Restrictions Checking for Heterogeneous Transportation Problem

M Osechkina¹, T Osechkina²,³

¹Tickvision, St. Petersburg, Russia
²Department of natural sciences and humanitarian education, St. Petersburg State Forestry University named after S.M. Kirov, St. Petersburg, Russia
³Higher School of Printing and Media Technologies, Saint-Petersburg State University of Industrial Technologies and Design, St. Petersburg, Russia

E-mail: mathschool_pstu@mail.ru

Abstract. Transportation costs optimization is very actual problem today because of logistic growth. In common case package type, origin and destination may impose restrictions on the delivery type and vehicle type for transporting. In some cases there may exist two packages, first of which can be delivered by first vehicle type and can’t be delivered by second vehicle type. Otherwise second package can be delivered by second vehicle type and can’t be delivered by first vehicle type. Or in some cases there are several vehicles types in fleet with different capacity and different size or different driver working time, so it is restriction for transport problem. We offer a heuristic algorithm for solving transport problems with heterogeneous vehicle fleet. The algorithm is based on the extension of Clark-Wright method, which is used to solve transport problem.

1. Introduction

Minimization of transportation total cost subject to certain restrictions is one of the central problems of logistics. So it is quite understandable that there is a desire to reduce transportation cost, that is price of fuel, working time of drivers, size of vehicle fleet and so on.

Input of the transportation problem is a depot or a starting point, several customers with addresses to be visited to collect or leave package, distances directed graph with vertices corresponding to these addresses, as well as at the depots, and set of vehicles. It is necessary to deliver packages by this set of vehicles, so that the sum length of all used graph edges is minimal, or in some cases so that the elapsed time is minimal, while respecting the limitations. In most cases, the packages must be picked up from the depot or delivered to the depot, but some travel companies offer the service of delivery of connected packages, i.e. the need to pick up package from one client on the same route to deliver to another, without going to the depot between them).

Since the graph distances is directional, two vertices are connected by two links in both directions, which is useful, for example, if the traffic on some part of the route is one-sided, and the distance from one point to another is not equal to the distance of return. Assuming that the vehicle is moving uniformly to all sections of the road, the calculation of optimal time path is reduced to finding an optimal path according to the distance. But in cases where the vehicle is moving unevenly, which in reality happens often because of traffic jams, road quality differences and speed limits, minimization...
of route total elapsed time is not equivalent to minimization of the total distance. If for problem decision it is fundamental to calculate the optimal solution by time criterion we should find the average speed by statistics for each edge of graph associated to route section, and then we should construct graph, in which the weights are the predicted time and not the distance.

Typically, in transportation problem there are capacity restrictions of the vehicle weight and volume, in general vehicles may vary according to these parameters. Each package respectively have the weight and volume. Often it is necessary to solve the transportation problem with time windows: the packages must be collected in a certain time period if the vehicle arrived before time window begin the driver have to wait for the requested time. In case of transportation problem with time windows the time of package service is introduced too, that is time of package unloading or loading. In addition to these restrictions, sometimes it becomes necessary to introduce a limit for the service time of the vehicle (which can also vary for different vehicles, for example, if the drivers work in turn), the type of package and the type of vehicle load (vertical, horizontal, for bulk good and so on), the ability of vehicle entry to customers address and others. Any of these limitations obviously makes more complicate the calculation of solutions as in addition it is necessary to construct routes that could be assigned to given vehicles with very different restrictions. If all vehicles are equal in capacity and other restrictions, then the transportation problem is homogeneous, otherwise it is heterogeneous.

The transportation problem with restrictions is np-hard task, that is the complexity of calculation is growing exponentially as a function of the number of packages or graph vertices. Usually in case of a large number of packages it is sufficient to find an approximate solution that satisfies the constraints and the total length of all routes is not much different from the optimum. In such cases, we can use heuristic algorithms with polynomial complexity to save solution calculation time. We offer a heuristic algorithm for solving the transportation problem with heterogeneous restrictions for vehicles, which can have different capacity, load time, work time, ability of entry to the addresses zone etc. Besides in the described below algorithm the possibility of conveying the connected packages is allowed.

2. Existing solutions
The capacity transportation problem can be homogeneous or heterogeneous with restrictions: in a homogeneous variant vehicles have the same capacity in a heterogeneous maximum volume or allowable weight is various.

Since the transportation problem is broadly applicable and important in practice there are many precise and heuristic methods to solve it. Among the precise methods for the solution most commonly used method of branch and cut method, branch and bound method, the Lagrange method. The authors of [4] tested the algorithm using Lagrangian method on a graph with 25 vertices, which are 24 customer and depot. The authors of [7] propose the exact algorithm for solving the heterogeneous capacity transportation problem that is based on the LP-relaxation and Lagrangian relaxation. The authors of [8] propose an algorithm based on cut and bound method for heterogeneous transportation problem with a different capacity and the cost of vehicles. As mentioned by the authors of [5] algorithm branch and bound runs unacceptably slow when a large number of customers (or vertices). Algorithm branch and cut work out for the time allowed if the number of clients is less than 100. The exact methods are limited in the performance, as is the transportation problem is NP-hard.

Genetic algorithm [13], the Clarke-Wright method [6], the search method with restrictions (tabu search method), the Monte Carlo method combined with the BGV [5], the methods of clustering, followed by the construction of the route for each cluster [3], [14] are heuristics methods of transportation problem solution. The authors of [9] offer tabu search method for a heterogeneous capacity transportation problem on the assumption that the number of each type of vehicle is not limited but in the real world it is too strong assumption. In [10] algorithm based on tabu search for transportation problem with heterogeneous constraints is proposed, the number of vehicles of each type is given in advance. First of all homogeneous problem for the transportation of each type is solved, and then on the basis of the solutions new solution is built. The authors [11] suggest a heterogeneous solution of the transportation problem by using a genetic algorithm.
3. Formalization

Let $I = i_0, i_1, i_2, \ldots, i_n$ is a customer set (i is route index), one map is given for some (or perhaps no one) customers and their connected packages: $F: I \rightarrow I$. In this map the argument the departure customer (graph vertice) and the value is delivery customer (graph vertice).

- $Cl = c_{i_0}, c_{i_1}, c_{i_2}, \ldots, c_{i_n}$ - the weight of each package;
- $VI = v_{i_0}, v_{i_1}, v_{i_2}, \ldots, v_{i_n}$ - the volume of each package;
- $TI = t_{i_0}, t_{i_1}, t_{i_2}, \ldots, t_{i_n}$ - the type of package for each customer, which can be one of those values: delivery from depot, delivery to the customer, the departure from customer (the same package have different type for departure and arrival point);
- $ZI = z_{i_0}, z_{i_1}, z_{i_2}, \ldots, z_{i_n}$ - zones of each customer (graph vertice). For convenience, validation assignment route vehicle customer zone will be stored as a sequence of 0 and 1 of fixed size $k$, where $k$ is the total number of zones. Notice, that $z_{ij}[p - 1] = 1$ if customer $i_j$ is situated in zone with index $p$, 0 otherwise.

With each route we associate a sequence of 0 and 1, which indicates which zones have been intersected by this route. Let $s$ - this sequence, then the $s[p] = ztr_1 \lor ztr_2 \lor \ldots \lor ztr_n$, that is the result of a bitwise OR for all zones sequences corresponding to all customers from this route.

$TW = tw_{i_0}, tw_{i_1}, tw_{i_2}, \ldots, tw_{i_n}$ is the customer time windows, each of which is minimum and maximum allowable time of arrival and customer service time.

$G(I, E)$ - graph of roads connecting each customer with each customer and depot.

$c_{ij}$ is weight edge between $i$ and $j$ vertives. Optional

$c_{ij} = c_{ji}$;

$T = t_{i_0}, t_{i_1}, t_{i_2}, \ldots, t_{i_n}$ is set of vehicles;

$CT = c_{t_0}, c_{t_1}, c_{t_2}, \ldots, c_{t_m}$ is maximum allowable weight for corresponding vehicle;

$VT = v_{t_0}, v_{t_1}, v_{t_2}, \ldots, v_{t_m}$ is maximum allowable volume for corresponding vehicle;

$zt = zt_{i_0}, zt_{i_1}, zt_{i_2}, \ldots, zt_{i_m}$ is sequence of 0 and 1, which corresponds to the zones which are allowed for vehicle;

$zt_{ij}[p] = 1$ if the vehicle $t_j$ is allowed zone $p$; 0 otherwise;

$TWT = twt_{0}, twt_{1}, twt_{2}, \ldots, twt_{m}$ - a set of time windows of vehicle working time, that is, the departure time from the depot and the time that vehicle should return to the depo.

4. Checking routes set to the vehicles

Usually rewording constraints strongly affects the algorithm. Note in the heuristic algorithms set of routes is often initialized first, and then set is subsequently improved. Initial set of routes may not satisfy the constraints of vehicles set. If the constraints are homogeneous then checking the routes on each algorithm iteration is not difficult: the route is approved if and only if it satisfies the uniform restrictions. But if the restrictions on vehicles are vary checking of routes set restrictions is complicated. Initially the number of routes may exceed the number of available vehicles, so it is necessary to formulate a criterion for which routes set is not rejected if routes of this set can not be spread over the vehicles at this stage, but route set will be rejected, if after any subsequent treatment this routes set can not be spread over vehicles. Note since the transportation problem is NP-hard it’s not necessary to guarantee that heuristic algorithms solution is exact. Therefore it is sufficient to formulate condition, which allows to get not the exact solution but solution with some good enough precision. Let’s describe the algorithm of routes’ set restrictions checking.

1) For each route we find all vehicles, such that the route satisfies restrictions related to vehicle. For example, the zones verification is passed if and only if $sr \land st = sr$, where $sr$ is a sequence of 0 and 1 for the zones of the route, and the $st$ is the sequence of 0 and 1 for vehicle zones, & - Bitwise AND.

Checking of weight capacity is succeed if and only if total weight of all route packages at any moment of transportation through the route is less or equal the maximum allowable weight capacity of the vehicle. Similarly volume capacity should be checked.
2) Let sort routes the following way: if the number of associated with first route vehicles is less than the number of associated with second route vehicles then the first route is less that the second. If the vehicles number is equal for both route then let sort the associated by route vehicles on the identifiers and compare the array of identifiers by element. If the first array is less than the second then the first route is less than the second.

3) We consider a subset of the routes constructed in the following way: from the sorted set of routes we select the first \( k \) elements, where \( k \) varies from 1 to the total number of routes. Now, for each such subset let’s construct a bipartite graph, where one part is a subset of the routes from subset, and the second part is a set of all vehicles. Let’s connect vertices from parts by edge if and only if vehicle satisfies the route restrictions. We calculate the maximum matching for each graph and construct array of maximum matching cardinalities that are associated with subsets of routes (approx. The order of the elements in the constructed array corresponds to increment the number of elements in subsets of routes).

4) Let’s carry out the iteration algorithm for solving the transportation problem for the Advancement of multiple routes, we get a new set of routes. Carry out steps 1-3 for new set of routes, and then compare arrays of maximum matchings \( M_1 \) and \( M_2 \) for two sets. If there is a number \( i \) such that for all \( j < i \), \( M_1[i] = M_2[j] \), and the \( M_1[i] > M_2[j] \), then the combined route is spoiled and the set is rolled back to the previous iteration, or an improvement of the set of routes is considered successful.

5. Clarke and wright algorithm modification for connected packages

Let us review the essence of Clarke and Wright algorithm [6]. On the route initialization stage, we assume that each client is serviced by single vehicle, So we construct only the routes of depot-client-depot type. Then we create a list of savings for couples of clients \( v_i v_j - s_{ij} \), which is defined as the distance savings, which will be saved, if two clients will be served by one vehicle one right after the other, and will not be served by each of his vehicle:

\[
s_{ij} = c_{0i} + c_{0j} - c_{ij}.
\]

By triangles inequality \( s_{ij} > 0 \). Savings list is ordered in descending order. Let us each time choose the greatest saving, and if its corresponding customers are first or last vertices in related route, we try to merge the respective routes "customer-to-customer". If a new set of routes passes for vehicles restrictions, we save new routes set with merged route. So we pass around the savings list, merging routes if it is possible.

Usually in the transportation problem it is assumed that packages must be picked up either from the depot and taken to the depot, or vice versa to be taken away from the customer to be delivered to the depot. Clarke-Wright algorithm is designed to work precisely with such a formulation, there are no customers (graph vertices) that should be added into the same route in a predetermined order. But it so happens that the packages should be delivered from one customer to another, so there are two points - vertices that should be assigned to one route. In this case, instead of the proposed radial routes (depot - customer - depot) during initialization of routes for connected packages let create triangle route (depot - customer-sender - customer-recipient - depot). We offer to calculate savings of routes merging as follows: \( AB \) - link to the route, which will be "inserted into a second route", \( CD \) - the first and last point of the second path, respectively, excluding the depot, O is the depot. The saving is calculated as saving = \( AB + OC + DO - AC - DB \). Note that the direction of the route is meaningful, so the attribute is invertible is introduced for each route, which is set to true, if the route does not contain connected by package customers, and false if it does. Thus, we calculate value for all saving pairs “Space to insert in the route - a different route." If route is invertible the addition gain insertion couples "place to insert in the route - other inverted route" are calculated. After the calculation of all savings, let choose the greatest saving and the corresponding pair of "position to insert - route to insert", then by selected a pair let construct a new route. If the routes set with the new route is passed checking of vehicles restriction, the saving set is updated as routes set, or delete the selected saving from the saving list. Updating the set of pairs and savings involves removing of all pairs corresponding to at least one of
the merged routes and related savings and adding pairs with new route and remaining routes, as well as calculating and adding respective savings. So we try to consistently merge routes for all non-negative savings.

Using Clarke-Wright algorithm we does not have to update the array of savings after routes merging. But in the proposed algorithm it is necessary to add new savings after the construction of new route, because routes can be merged not only "end-to-end", but also to inserted one into another route. It is obvious that new pair of vertices are formed while new route constructing, we can use this pair by merging other routes, and therefore savings set must be updated. It is useful to choose the structure of the set so that we can quickly select the maximum savings element at any time, including after the upgrade. In addition, it is desirable to speed up the renewal of the set, while adhering to the requirements of quick selecting the maximum element. For example, red-black trees can be chosen for such set storing [1], [2]. With this method of storing a plurality of access to the largest element may be carried over the O (1), inserting and deleting elements takes O (ln n), where n is the number of elements in the tree.

6. Experiment

For experiment data from VRPLIB [15] is used. Several samples with info about graph of distances between customers, packages weight, maximum vehicle capacity is used to run Clarke-Wright algorithm. All this local problems are homogeneous, that is all vehicles are equivalent. In the first table column there is used file name. In the second column there is maximum vehicle capacity. Number of calculated routes is in the third column and it is equal to number of vehicles in all samples. There is no routes without vehicle in these homogeneous problems samples. In forth column is final route length.

| File         | Vehicle capacity | Routes number | Sum length |
|--------------|------------------|---------------|------------|
| E101-08e.dat | 200              | 8             | 890.718    |
| E101-10c.dat | 200              | 10            | 830.584    |
| E101-14s.dat | 112              | 14            | 1146.01    |
| E121-07c.dat | 200              | 7             | 1064.25    |
| E135-07f.dat | 2210             | 7             | 1198.68    |
| E151-12b.dat | 200              | 12            | 1113.98    |
| E151-12c.dat | 200              | 12            | 1113.98    |
| E101C11r.dat| 2043             | 11            | 1434.07    |
| E131B14r..dat| 1928             | 14            | 2890.4     |

After we run homogeneous problems we construct data for heterogeneous problems. We combined data from pairs of file: vehicles sets, distances graph and packages were merged. So in such samples we get different types of vehicles with different capacity for one problem. After that generated problems was solved by Clarke-Wright algorithm. 45 heterogeneous problems was generated. We compared total route length for each of generates data with sum of route total length of merged homogeneous problem samples. So average relation of total length for merged data and sum of total length for separate associated data is equal to 0.940653. Maximum of such relations is 1.18142. Number of generated samples where it was impossible to assign vehicle for each route is 18. Average relation of total routes length for such samples is 1.02. Average number of routes without truck for all generated samples is 0.42. Note that in homogeneous case all vehicles was used. And in real world usually there is reserve vehicles. Additionally we made an experiment with emulation of distribution of packages by zones. With each file unique zone was associated. So customers associated to one file could be visited only by vehicle from the same file. And other customers was associated to other unique zones. Total length of all routes for generated problem conditions was always equal to sum of total length of routes that was constructed as solution of separate problems. Each route was successfully assigned to vehicle.
7. Conclusion
This article has provided a method of heuristic algorithms generalization for heterogeneous transportation problem. The algorithm provides the way to plan packages delivery from depot to customer or from customer to depot as well as from one customer to another.

References
[1] Kormen T, Charles Leiserson, Rivest R K Stein to algorithms 2nd ed. (M.: Publishing House "Williams") pp 336-364
[2] STL's Red-Black Tree Janson Shankel http://www.drdobbs.com/cpp/stls-red-black-trees/184410531?pno=1
[3] A Beam search Based Algorithm for the Capacitated Vehicle Routing Problem with Time Windows, Hakim Akeb, Adel Bouchakhchoukha, Mhand Hifi, Proceedings of the 2013 Federated Conference on Computer Science and Information Systems pp 329-336
[4] Acceleration of Lagrangian Method for the Vehicle Routing Problem with Time Windows, Hadi Karimi, Abbas Seifi International Journal of Industrial Engineering & Production Research vol 23 4 pp 309-315
[5] Frank W Takes, Walter A Kosters Applying Monte Carlo Techniques to the Capacitated Vehicle Routing Problem http://www.liacs.nl/~ftakes/pdf/vrp.pdf
[6] Clarke G and Wright J 1964 Scheduling of vehicles from a central depot to a number of delivery points Operations Research 12 pp 568-581
[7] Roberto Baldacci, Aristide Mingozzi A unified exact method for solving different classes of vehicle routing problems http://www.math.ucdenver.edu/~billups/courses/ma7025/papers/Baldacci Vehicle%202009.pdf
[8] A Robust Branch-Cut-and-Price Algorithm for the Heterogeneous Fleet Vehicle Routing Problem, Artur Pessoa, Eduardo Uchoa, Marcus Poggi de Aragao http://www.optimization-online.org/DB_FILE/2008/06/2004.pdf
[9] Gendreau M, Laporte G, Musaraganyi C and Taillard E D 1999 A tabu search heuristic for the heterogeneous fleet vehicle routing problem Computers and Operations Research 26 1153-1173
[10] Éric D Taillard 1996 A Heuristic Column Generation Method for the Heterogeneous Fleet VRP CRT-96-03
[11] Luis S Ochi, Dalessandro S Vianna, Lucia M A Drummond, Andre O Victor A Parallel Evolutionary Algorithm for the Vehicle Routing Problem with Heterogeneous Fleet http://ipdps.cc.gatech.edu/1998/biosp3/bispp5.pdf
[12] Chris Groer, Bruce Golden, Edward Wasil A Parallel Algorithm for the Vehicle Routing Problem http://web.ornl.gov/~tcg/publications/par_vrp.pdf
[13] Yaw Chang and Lin Chen 2007 Solve the Vehicle Routing Problem with Time Windows via a Genetic Algorithm Discrete and continuous dynamical systems supplement pp 240-249
[14] Suthikarnnarumai N 2008 A Sweep Algorithm for the Mix Fleet Vehicle Routing Problem Proceedings of the International MultiConference of Engineers and Computer Scientists vol II IMECS (Hong Kong)
[15] Daniele Vigo VRPLIB: A Vehicle Routing Problem LIBrary http://or.det.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html