Comparison of heuristic algorithms for solving a specific model of transportation problem

Cite as: AIP Conference Proceedings **2302**, 060004 (2020); https://doi.org/10.1063/5.0033505
Published Online: 03 December 2020

I. Georgiev, D. Grozev, V. Pavlov, and E. Veleva

Comparison of Heuristic Algorithms for Solving a Specific Model of Transportation Problem

I. Georgiev[1,a)], D. Grozev[2,b)], V. Pavlov[1,c)] and E. Veleva[1,d)]

[1]Dept of Applied Mathematics and Statistics, Angel Kanchev University of Ruse, 7017 Ruse, 8 Studentska str., Bulgaria
[2]Dept of Transport, Angel Kanchev University of Ruse, 7017 Ruse, 8 Studentska str., Bulgaria

[a)]irgeorgiev@uni-ruse.bg
[b)]dgrozev@uni-ruse.bg
[c)]Corresponding author: vpavlov@uni-ruse.bg
[d)]eveleva@uni-ruse.bg

Abstract. A specific transportation problem is presented in the paper. A commonly used variant for the cost value determination is analysed. A Matlab code for solving this specific transportation problem by the heuristic algorithms is developed The optimization model used is related to solving a problem with nondeterministic polynomial-time (NP) hardness and large dimension. A comparison of different heuristic algorithms for solving this problem is made. Results are commented.

Keywords: Transportation problem, mixed integer optimization, heuristic algorithms

INTRODUCTION

In the management of various companies, problems arise with the handling of machines, raw materials, finances, large groups of people and other resources related to transport work. Mathematical methods used to solve such kind of problems are called quantitative methods. One of them is the transportation problem.

It is important for every transport company owner to keep their expenditure low. Each courier has an unique set of services and prices, so it is important to compare prices to make sure they have taken the best possible solution for their business. In turn, customers generally do extensive research until they find a courier whose services and pricing fit their needs. Deciding on delivery transactions is a very important part of maintaining balanced accounts and satisfied customers.

There are many different ways to determine what delivery fee should be charged. The transport costs are usually based on the weight of one shipment and on the distance. If the standard rate is set too low, there is a risk of not profiting from orders. If a fixed delivery price is set too high, there is a risk of losing customers who do not want to pay for additional charges.

There are 11 courier companies with nearly 30 offices in the city of Ruse, Bulgaria [1]. Our analysis of the transportation problems is based on the information from transport companies working in the city of Ruse.

EXPOSITION

The classical transportation problem is to minimize function Z [2,3,4]:

$$min\, Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

subject to the following limitations:

$$\begin{aligned} &\sum_{j=1}^{n} x_{ij} = a_i, \ \ \forall i = \overline{1..m} \\ &\sum_{i=1}^{m} x_{ij} = b_j, \ \ \forall j = \overline{1..n} \\ &x_{ij} \geq 0, \ \ i = \overline{1..m}, \ \ j = \overline{1..n}. \end{aligned} \tag{2}$$

Here Z is the total transport cost; m is the number of sources A_i that offer quantities a_i. The destinations B_j are n in number and they are looking for quantity b_j. The transport cost for a single unit from the i-source to the j-destination is c_{ij} and x_{ij} is the unknown quantity of transport units from the i-source to the j-destination.

Application of Mathematics in Technical and Natural Sciences
AIP Conf. Proc. 2302, 060004-1–060004-7; https://doi.org/10.1063/5.0033505
Published by AIP Publishing. 978-0-7354-4036-4/$30.00

In practice, the classical transportation problem is limited in scope. The real problems that arise in the transport of goods are related to many more factors, some of which are of a random nature. To predict many of these random factors usually is very difficult. Also every company and enterprise has its own specific features and requirements. For this reason, it is impossible to build a general mathematical model and hence each specific problem must be modelled individually according to its specific characteristics and features.

Some of these difficulties are:

- Many companies in the transport and logistics sector want to provide a “constant” value per kilometre, independent of the volume of shipment. Short distance deliveries are not as profitable as the long ones and in order to compensate for lost earnings, such a price is formed. The cost C_{ij}^0 of passing a unit distance from the *i*-source to the *j*-destination usually depends on the distance K_{ij} between them. At smaller distances, it is usually bigger.
- The cost of the fuel for a vehicle depends on its load. The more loaded the vehicle, the higher the fuel consumption it has. The cost of the fuel for a vehicle at the maximum load may be double increased relative to the unloaded vehicle for the same route. For this, besides the “hard” price per unit distance C_{ij}^0, carriers are also interested in introducing a price C_{ij}^1 for each unit of load when traveling per unit distance.
- There are many other factors that can influence the pricing. These may be the type of goods carried, *i.e.*, the risk of cargo damages. Another such factor is the road – it matters whether it passes through many settlements where there are many speed limits, whether the road is mountainous or a highway. Different road networks have a different impact on transport time and consumption. Last but not least, climate conditions can also be questioned – in winter the vehicle’s consumption increases. Taking into account all the factors and constructing a common mathematical model is impossible not only because of its huge quantity, but also because some of these factors are random and unpredictable.

Our mathematical model, presented here is formed on the three main factors already listed above.

To determine the mathematical model of a transportation problem, the following indications must be entered:

$A_i -$ sources , $i=\overline{1..m}$;
$B_j -$ destinations, $j = \overline{1..n}$;
$C_{ij}^0 -$ the cost of passing a unit distance from the *i*-source to the *j*-destination;
$C_{ij}^1 -$ the cost per unit distance from the *i*th source to the *j*th destination, proportional to the unit load;
$K_{ij} -$ the distance from the *i*th source to the *j*th destination;
$X_{ij} -$ the quantity of transport units from the *i*-source to the *j*-destination;
$a_i -$ the amount or volume available to the *i*-source;
$b_j -$ the amount or volume needed to be delivered to the *j*-destination;
$Z -$ total cost.

With these notations the mathematical model can be rewritten as:

To minimize the function Z

$$minZ = \sum_{i=1}^{m}\sum_{j=1}^{n} Z_{ij} \tag{3}$$

under standard restrictions:

$$\begin{gathered}\sum_{j=1}^{n} X_{ij} = a_i,\ \ \forall i = \overline{1..m}\\ \sum_{i=1}^{m} X_{ij} = b_j,\ \ \forall j = \overline{1..n}\\ X_{ij} \geq 0,\ \ i = \overline{1..m},\ \ j = \overline{1..n}\end{gathered} \tag{4}$$

where

$$Z_{ij} = \begin{cases} K_{ij}\left(C_{ij}^0 + C_{ij}^1 X_{ij}\right), & X_{ij} > 0 \\ 0, & X_{ij} = 0 \end{cases}$$

i.e., if a transport from the *i*-source to the *j*-destination is made, then $Z_{ij} \neq 0$ increases linearly. Otherwise $Z_{ij} = 0$.

The specificity here is in the discontinuity of the components of the total cost *Z* (Figure 1). Due to the discontinuity of Z_{ij}, the direct application of the linear programming is not possible. To overcome this feature, for each variable X_{ij} binary variables y_{ij} are introduced:

$$y_{ij} = \begin{cases} 1 & if\ X_{ij} > 0 \\ 0 & if\ X_{ij} = 0 \end{cases}.$$

Problem (3) is then reformulated as follows:

$$\min Z = \sum_i \sum_j K_{ij}(C_{ij}^0 y_{ij} + C_{ij}^1 X_{ij}). \quad (5)$$

Restrictions (4) are retained, but additional limitations are imposed on the binary variables:

$$X_{ij} \leq M y_{ij},\ i = \overline{1,..m},\ j = \overline{1,..n} \quad (6)$$

where M is a very large number (M>>1).

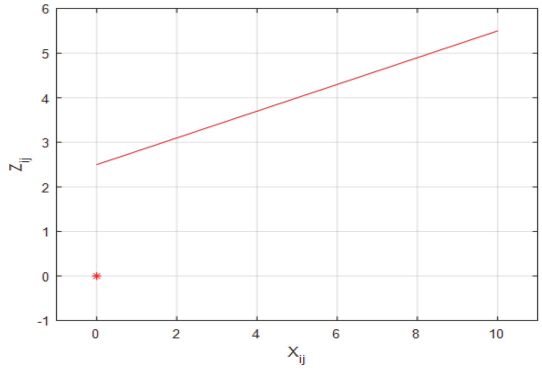

FIGURE 1. Graph of the expenditure component Z_{ij}

The transportation problem (4)-(6) has the following features, namely it is

- Mixed-integer problem of linear optimization;
- The problem is likely to become large in size.

It is known that the mixed-integer problems are generally NP-complete (nondeterministic polynomial-time) problems and their exact solution is labor-intensive and requires a large amount of computing resource and time. Even with a relatively small number of sources and destinations, once the binary variables have been entered, the size of the problem is doubled. Such types of problems are common in the logistics of all modes of transport. They solve many questions related to attachments, timetables, volume transport, and many more (see [5,6]).

The following solutions are possible:

- Classic - Branch and Bound and Gomori Methods. They are suitable for small-scale problems. Their advantage is that they give the exact solution. Large problems are inappropriate, because it can take unacceptably long calculation time and a large memory resource ([2, 4]).
- Heuristic and Probabilistic Approaches [7,8,9]. The development of these approaches over the last few years has made it possible to solve real problems with large dimensions. Typical of these algorithms is that they are much faster than the classic ones. Like any heuristic or probabilistic algorithm, the solution is not always optimal, but in most cases it is satisfactory enough [10, 11].
- Neural Networks and Artificial Intelligence [12]. Recently, neural networks and artificial intelligence have evolved. More and more problems have become available to solve thanks to this development. Mathematical optimization in this respect is no exception. Many optimization problems are solved through them. Here, as with heuristic algorithms, a good enough solution is sought rather than the exact one. This, of course, is at the expense of the smaller computational resource and time.

Software Matlab provides a great choice of heuristic approaches to solving mixed-integer problems [13]. Using them, a file-function for solving the problem (4)-(6) is realized. The input and output data are as follows:

C_0 - matrix with constant costs per km, independent of volume;
C_1 - matrix with costs per unit volume per unit km;
KM - matrix of distances between cities;
a - vector with volumes of “suppliers”;
b - vector with volumes of “users”;
Z - minimum price for the overall solution;
X - matrix of the quantities transported (the solution of the task).

In case of an unbalanced transportation problem, the program automatically balances it. When solving a blocked transportation problem, the corresponding element in the matrix KM of the distances is assigned a large enough number, of the order of 10^9.

The program allows one to choose the heuristic method by which the solution is sought [13]. In the program code in “options”, one can choose any of these algorithms and put them in the place of the *basic*.

```
function [Z, X] = trans_nlin1h (C0, C1, KM, a, b)
% M- Very large number. It comes from the model.
% C0 - matrix with constant costs per km, independent of volume
% C1 - matrix with costs per unit volume per unit km
% KM - matrix of distances between cities
% a - vector with volumes of "suppliers"
% b - vector with volumes of "users"
% Z- minimum price for the overall solution
% X - quantity matrix (solution of the problem)

C0=C0.*KM;
C1=C1.*KM;
[m,n]=size(C0);
a=a(:);
b=b(:);
suma=sum(a);
sumb=sum(b);
Iflag=0;
if sumb>suma
   Iflag=1;
   nulb=zeros(1,length(b));
   C0=[C0;nulb];
   C1=[C1;nulb];
   a=[a;sumb-suma];
end
if sumb<suma
   Iflag=2;
   nula=zeros(length(a),1);
   C0=[C0,nula];
   C1=[C1,nula];
   b=[b;suma-sumb];
end
B=[a;b];
M=max(B);
[m,n]=size(C1);
f1=reshape(C1',m*n,1);
f0=reshape(C0',m*n,1);
f=[f1;f0];
A1=zeros(m,m*n);
A2=eye(n);
A3=eye(n);
for k=1:m
   A1(k,((k-1)*n+1):(k*n))=1;
end
for k=1:(m-1)
   A2=[A2,A3];
end
A=[A1;A2];
A=[A,zeros(m+n,m*n)];
lb=zeros(2*m*n,1);
ub=[M*ones(m*n,1);ones(m*n,1)];
s=m*n+1:2*m*n;s=s';
Aeq=[eye(m*n),-M*eye(m*n)];
Beq=zeros(m*n,1);
opt=optimoptions('intlinprog','Heuristics','basic');
  %'basic'
  %'intermediate'
  % 'advanced'
  % 'rss'
  % 'rins'
  % 'round'
  % 'diving'
  % 'rss-diving'
  % 'rins-diving'
  % 'round-diving'
  % 'none'
[X,Z]=intlinprog(f,s,Aeq,Beq,A,B,lb,ub,[],opt);
X=X(1:n*m);
X=reshape(X,n,m);X=X';
if Iflag==1
   X(end,:)=[];
elseif Iflag==2
   X(:,end)=[];
end
Z=round(Z,9);
X=round(X,9);
```

The latest versions of Matlab [13] have a large set of heuristic techniques for calculating mixed-integer problems. At present, these techniques, with a brief description, are (see [7]):

- 'basic' (default) [13] — Runs 'round', then 'rss'. The solver does not run later heuristics when earlier heuristics lead to a sufficiently good integer-feasible solution.
- 'intermediate' [13] — First runs 'round', then 'rins', then 'rss'. The solver does not run later heuristics when earlier heuristics lead to a sufficiently good integer-feasible solution.
- 'advanced' [13] — First runs 'round', then 'diving', then 'rins', then 'rss'. The solver does not run later heuristics when earlier heuristics lead to a sufficiently good integer-feasible solution. The solver uses only the fractional diving and guided diving heuristics for 'diving'.
- 'rins' [13] — intlinprog searches the neighborhood of the current best integer-feasible solution point (if available) to find a new and better solution.
- 'rss' [13] — intlinprog applies a hybrid procedure combining ideas from 'rins' and local branching to search for integer-feasible solutions.
- 'round' [13] — intlinprog takes the linear problem solution to the relaxed problem at a node. It rounds the integer components in a way that attempts to maintain feasibility.

- 'diving' [13]— intlinprog uses heuristics that are similar to branch-and-bound steps, but follow just one branch of the tree down, without creating the other branches. This single branch leads to a fast "dive" down the tree fragment, hence the name "diving." Currently, intlinprog uses six diving heuristics in this order, until it obtains an integer-feasible point with a relative gap of less than 5% or takes too much time:
- Vector length diving
- Coefficient diving
- Fractional diving
- Pseudo cost diving
- Line search diving
- Guided diving (applies when intlinprog already found at least one integer-feasible point)

Diving heuristics generally select one variable that is supposed to be integer-valued, for which the current solution is fractional. They then introduce a bound that forces that variable to be integer-valued, and solve the associated relaxed linear problem again. The method of choosing the variable to bound is the main difference between the diving heuristics [10,13]:

- 'rss-diving' or 'rins-diving' — intlinprog tries 'diving' first, then (if necessary) the named heuristic method ('rins' or 'rss').
- 'round-diving' — intlinprog tries 'round' first, then (if necessary) tries 'diving'.

NUMERICAL EXPERIMENTS

First let the following data are given:

$$a = (2000; 1000) \;\; b = (1100,1500,900)$$

$$K_{ij} = \begin{pmatrix} 195 & 285 & 310 \\ 50 & 405 & 480 \end{pmatrix} \qquad C_{ij}^0 = \begin{pmatrix} 0.81 & 0.75 & 0.74 \\ 0.85 & 0.72 & 0.70 \end{pmatrix} \qquad C_{ij}^1 = \begin{pmatrix} 0.0702 & 0.0612 & 0.0587 \\ 0.0847 & 0.0492 & 0.0417 \end{pmatrix}$$

Matlab solution using a **heuristic approach** (optimoptions('intlinprog','Heuristics','basic')) is:

```
>> C0=[0.81 0.75 0.74;0.85 0.72 0.7];C1=[0.0702 0.0612 0.0587;0.0847 0.0492 0.0417];
>> KM=[195 285 310;50 405 480];a=[2000;1000];b=[1100 1500 900];
>> format long g
>> tic,[Z,X]=trans_nlin1h(C0,C1,KM,a,b), toc
LP:            Optimal objective value is 39045.700470.
```

TABLE 1. Comparison of times in sec. for calculating problems of varying size using the classical Branch and Bound and heuristic methods

Size of the problem after balancing and adding binary variables	32	40	60	200	800	1800
Classical Branch and Bound method in sec.	13.7659	145.8208	2100.4192	-	-	-
Heuristics: 'basic' in sec.	0.0760	0.0781	0.0586	0.0605	4.4012	105.7407
Heuristics:'intermediate' in sec.	0.0303	0.0333	0.0323	0.0517	4.3408	104.0006
Heuristics: 'advanced' in sec.	0.0370	0.0636	0.0320	0.0593	4.3337	104.3405
Heuristics: 'rss' in sec.	0.0235	0.0273	0.0377	0.0569	4.3746	110.2070
Heuristics: 'rins' in sec.	0.0309	0.0406	0.0299	0.0600	4.3340	107.5775
Heuristics:'round' in sec.	0.0344	0.0358	0.0372	0.0639	4.3720	105.6553
Heuristics: 'diving' in sec.	0.0384	0.0384	0.0316	0.0546	4.3222	110.5756
Heuristics: 'rss-diving' in sec.	0.0253	0.0292	0.0271	0.0496	4.3381	105.5536
Heuristics: 'rins-diving' in sec.	0.0296	0.0330	0.0334	0.0523	4.3374	107.1016
Heuristics: 'round-diving' in sec.	0.0313	0.0323	0.0340	0.0464	4.3210	107.3457

Optimal solution is found. Intlinprog stopped at the root node because the objective value is within a gap tolerance of the optimal value, options. AbsoluteGapTolerance = 0 (the default value). The intcon variables are integer within the tolerance options. IntegerTolerance = 1e-05 (the default value). Next the optimal solution for the first case is written

```
Z =39045.70047
X =100     1500     400
     1000      0       0
```

Elapsed time is 0.042523 seconds. It can be seen that the solver has managed in less than 0.05 seconds. The size of the task after balancing is 18 variables, 9 of which are binary.

Solution in Matlab, using **Branch and Bound** ([14,15,16]) without using a heuristic approach is

```
>> tic,[Z,X]=trans_nlin1(C0,C1,KM,a,b), toc
Z = 39689.3
X =100     1500     400
    1000      0       0
```

Elapsed time is 0.710173 seconds. It can be seen that without the use of a heuristic technique the equation time has grown about 17 times.

Table 1 compares the times of calculation with the classical Branch and Bound method and some heuristic methods on tasks of varying size.

From the table we see that when using the classical Branch and Bound method without additional techniques, times become unacceptably large even with relatively small size of the problem. On the other hand, different heuristic approaches with the same dimension of the problem, require approximately the same and much smaller times. Even at a large scale the time is much more acceptable than this of the classic Branch and Bound method.

CONCLUSIONS

The advantage of transportation problem is that it can be applied in different areas related to the transport of goods and, if necessary, allow the model to be modified to adapt to specific situations and factors. The analysis of the work of logistics companies in the city of Ruse shows that there are many heterogeneous problems for which a common mathematical model is impossible to implement. There are a number of problems that are observed in the vast majority of them.

The mathematical model considered here describes a common problem in practice – the minimization of total transport costs, assuming additional requirements related to the cost of passing a unit distance, proportional to the unit load. The model leads to a mixed-integer linear optimization problem that requires a large amount of computing memory and time.

The implemented program in Matlab environment enables different heuristic approaches for transportation problem to be chosen. A comparison is made between the Branch and Bound method and different heuristic approaches. It was found that at a size above 60 variables, the Branch and Bound method gives unacceptable time for the solution. On the other hand, the considered heuristic methods even in the case of larger dimensions (about 1800 variables and more) give a solution in less than 2 minutes.

Heuristic approaches are much more time-efficient, although they do not always give the optimal solution but one close enough to it (3-5% difference from the optimal).

ACKNOWLEDGEMENTS

This paper contains results of the work on project No 2020 - FNSE – 04, financed by Scientific Research Fund of Ruse University, Bulgaria.

REFERENCES

1. Catalog of Transport and Logistic Companies, `https://transportnaborsa.cargoagent.net/katalog_na_transportni_firmi?id=&name=&user=&phone=&countrycode=BG&geoid=727523&zip=7000&city=7000%2C+Ruse&region=&category=3&submit=%D0%A2%D1%8A%D1%80%D1%81%D0%B8#offers`
2. H.A. Taha, *Operations Research, An Introduction* (2001, 2007).
3. N. Stefan, S. Oliver, and W. Karl-Heinz *Operations Research* (Springer-Verlag, Berlin Heidelberg, 2014), Softcover ISBN 978-3-642-54367-8, Series ISSN 0937-7433.
4. F.S. Hillier and G. J. Lieberman, *Introduction To Operations Research*, 10th edn (2015), ISBN 978-0-07-352345-3, MHID 0-07-352345-3.

5. M. Oliskevych (2018) Optimization of periodic unitary online schedule of transport tasks of highway road trains, *International Scientific Journal "Transport Problems"(Katovice, Poland)* **13**(1), 111–122.
6. V. Todorov, N. Ikonomov, S. Apostolov, I. Dimov, R. Georgieva, and Y. Dimitrov, "An improved "Walk on Equations" Monte Carlo algorithm for linear algebraic systems," in *Recent Advances in Computational Optimization* (Springer, Cham, 2020), pp. 215–236.
7. T. Berthold, *Primal Heuristics for Mixed Integer Programs* (Technischen Universität, Berlin, September 2006), available at `https://www.zib.de/groetschel/students/Diplom-Berthold.pdf`.
8. V. Todorov, I. Dimov, Tz. Ostromsky *et al* (2020) Advanced stochastic approaches for Sobol' sensitivity indices evaluation, *Neural Comput. & Applic.*, `https://doi.org/10.1007/s00521-020-05074-4`.
9. C. Wilbaut and S. Hanafi (2009) New convergent heuristics for 0–1 mixed integer programming, *European Journal of Operational Research* **195**(1), 62–74.
10. A. Asenov, V. Pencheva, and E. Savev (2011) Transport serving of the distribution in city, *International Scientific Journal "Transport Problems" (Katovice, Poland)*, 242–248, ISSN 1896-0596.
11. V. Todorov and I. Dimov, "Efficient stochastic approaches for multidimensional integrals in Bayesian statistics," in *International Conference on Large-Scale Scientific Computing* (pp. 454-462). (Springer, Cham, 2019), pp. 454–462.
12. Matlab documentation, `https://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html#bt6n8vs`.
13. E. Danna, E. Rothberg, and C. Le Pape (2005) Exploring relaxation induced neighborhoods to improve MIP solutions, *Mathematical Programming* **102**(1), 71–90.
14. V. Pavlov and I. Georgiev, *Optimization Methods with MATLAB* (Avangard Print, Ruse, 2016). [in Bulgarian]
15. V. Pavlov and I. Georgiev (2016) Branch-and-Bound Technique in Economics, *Journal of International Scientific Publications: Economy and Business* **6**, 34–42.