Maritime Technology and Research

https://so04.tci-thaijo.org/index.php/MTR

Research Article

Shipping patterns may be altered according to available refining capacities

Badri Toppur*

Rajalakshmi School of Business, Chennai, India

Article information

Received: July 20, 2021
Revised: September 8, 2021
Accepted: September 14, 2021

Keywords

Crude oil logistics,
Shipping patterns,
Mixed Integer Linear Programming (MILP),
Alternate optimal solutions

Abstract

Alternate optimal solutions in the mathematical programming solution of a real-world problem are rare, unless there are multiple objective functions. In a recent application of two mathematical programming solvers to a problem of crude oil logistics, I obtained two different optimal solutions. The almost optimal solutions obtained from the convex combinations of the two optimal solutions are feasible as real transportation choices if the optimal solutions cannot be implemented. It was observed that the two solutions led to different capacity utilizations at refineries downstream, and so did the many solution alternatives. This is useful information fora petroleum company during downtime for maintenance or capacity expansion.

1. Introduction

About six years ago, a logistics problem routinely faced by the Indian Oil Company was posed by the technical manager of one of the refineries to the author. The problem was soon formulated as a classical transshipment problem with supply nodes, transshipment nodes, and demand nodes. The crude oil was of 2 types- sweet crude and sour crude. Various ship sizes used in transportation were specified, such as Very Large Crude Carrier (VLCC), Suezmax, and Long Range (LR), along with their freight charges per voyage. These details are shown in **Tables 1** and **2**. The crude oil was received and downloaded to tank farms at two ports on the west coast of India. Details about two pipelines running from the two ports to three refineries were also provided. The refineries had differing capacities for processing the sour crude, and the refinery margins of the two products were also different.

Table 1 Vessel capacities.

VLCC: 270 thousand metric tons (TMT) capacity or 2 million bbl
Suezmax: 135 TMT capacity or 1 million bbl
Long Range (LR): 60 TMT capacity or 0.45 million bbl

*Corresponding author: Rajalakshmi School of Business, Chennai, India
E-mail address: badri.toppur@rsb.edu.in

Table 2 Freight charges for various ships.

VLCC from Persian Gulf costs \$2.00 million per voyage
Suezmax from Persian Gulf costs \$1.5 million per voyage
VLCC from West Africa costs \$3.5 million per voyage
Suezmax from North Africa and West Asia costs \$2.0 million per voyage
LR from Bombay High costs \$0.4 million per voyage

The model was initially implemented in a spreadsheet and solved to optimality. The objective function and 25 related constraints are given below this paragraph. The definitions of all the variables used in the model are in the Appendix. The optimal routes and flows are as indicated in **Figure 1**. A gross total of 35,595 TMT of crude oil was purchased from the production sites. This was about 405 TMT less than all the available supply from the production sites of 36,000 TMT. 19,580 TMT of high-grade gasoline and 14,236 TMT of low-grade gasoline were produced at the three refineries. The maximum gross profit was \$895,572,538. A general formulation is given in the paper, along with other details (Toppur & Sanyal, 2020).

Maximize:

$$\$58,400G_1+ \$47,450G_2-\$2000,000V_1-\$2000,000V_2-\$3,500,000V_3-\$3,500,000V_4 -\$1,500,000S_1-\$1,500,000S_2-\$2,000,00S_3-\$2,000,000S_4-\$400,000LR_1-\$400,000LR_2 -\$10851LV_{1,r}-\$17694LV_{2,r}-\$31160LV_{3,r}-\$10851HV_{1,r}-\$27694HV_{2,r}-\$31160HV_{3,r} -\$27157LM_{3,r}-\$30623LM_{2,r}-\$47466LM_{1,r}-\$27157HM_{3,r}-\$30623HM_{2,r}-\$47466HM_{1,r} -\$8139LV_{1,p}-\$20771LV_{2,p}-\$23370LV_{3,p}-\$8139HV_{1,p}-\$20771HV_{2,p}-\$23370HV_{3,p} -\$20368LM_{3,p}-\$20368HM_{3,p}$$

Subject to:

$$270V_1+ 270V_2+ 135S_1+ 135S_2 <= 9000 \quad (1)$$
$$270V_3+ 270V_4 <= 17000 \quad (2)$$
$$135S_3+ 135S_4 <= 2000 \quad (3)$$
$$60LR_1+ 60LR_2 <= 2000 \quad (4)$$
$$270V_1+135S_1+270V_3+135S_3+60LR_1 <= 30000 \quad (5)$$
$$270V_2+270V_4+135S_2+135S_4+60LR_2<= 9000 \quad (6)$$

Refining capacity constraints:

$$LV_{1,}+LM_{1,r}+LV_{1,p}+LM_{1,p} <= 15000 \quad (7)$$
$$HV_{1,}+HM_{1,r}+HV_{1,p}+HM_{1,p} <= 7500 \quad (8)$$
$$LV_{1,}+LM_{1,r}+LV_{1,p}+LM_{1,p}+HV_{1,r}+HM_{1,r}+HV_{1,p}+HM_{1,p} <= 15000 \quad (9)$$
$$LV_{2,}+LM_{2,r}+LV_{2,p}+LM_{2,p}\leq 15,000 \quad (10)$$
$$HV_{2,}+HM_{2,r}+HV_{2,p}+HM_{2,p}\leq 5400 \quad (11)$$
$$LV_{2,}+LM_{2,r}+LV_{2,p}+LM_{2,p}+HV_{2,r}+HM_{2,r}+HV_{2,p}+HM_{2,p} <= 9000 \quad (12)$$
$$LV_{3,}+LM_{3,r}+LV_{3,p}+LM_{3,p} <= 15000 \quad (13)$$
$$HV_{3,}+HM_{3,r}+HV_{3,p}+HM_{3,p} <= 13500 \quad (14)$$
$$LV_{3,}+LM_{3,r}+LV_{3,p}+LM_{3,p}+HV_{3,r}+HM_{3,r}+HV_{3,p}+HM_{3,p} <= 15000 \quad (15)$$

Flow conservation constraints:

$270V_1+135S_1-HV=0$ (16)

$270V_2+135S_2-HM=0$ (17)

$270V_3+135S_3+60LR_1-LV=0$ (18)

$270V_4+135S_4+60LR_2-LM = 0$ (19)

$LV-LV_{1,}-LV_{1,p}-LV_{2,r}-LV_{2,p}-LV_{3,r}-LV_{3,p} = 0$ (20)

$LM-LM_{1,}-LM_{1,p}-LM_{2,r}-LM_{2,p}-LM_{3,r}-LM_{3,p} = 0$ (21)

$HV-HV_{1,}-HV_{1,p}-HV_{2,r}-HV_{2,p}-HV_{3,r}-HV_{3,p} = 0$ (22)

$HM-HV_{1,}-HV_{1,p}-HV_{2,r}-HV_{2,p}-HV_{3,r}-HV_{3,p} = 0$ (23)

Blending constraints:

$G1-0.95LV-0.95LM=0$ (24)

$G2-0.95HV-0.95HM=0$ (25)

The next year, I obtained the results for the same formulation by using the *Gurobi* solver, which is the fastest and most advanced solver at present. I obtained the same optimal objective value as the spreadsheet, but a different solution vector of flow variables on the arcs of the network. This was a non-inferior or Pareto optimal solution, as is seen in multi-objective optimization problems. Indeed, there is a profit maximisation and a cost minimisation component in the same objective function, which contributes to this behaviour. In the following sections, I explore the managerial implications of having Pareto optimal solutions.

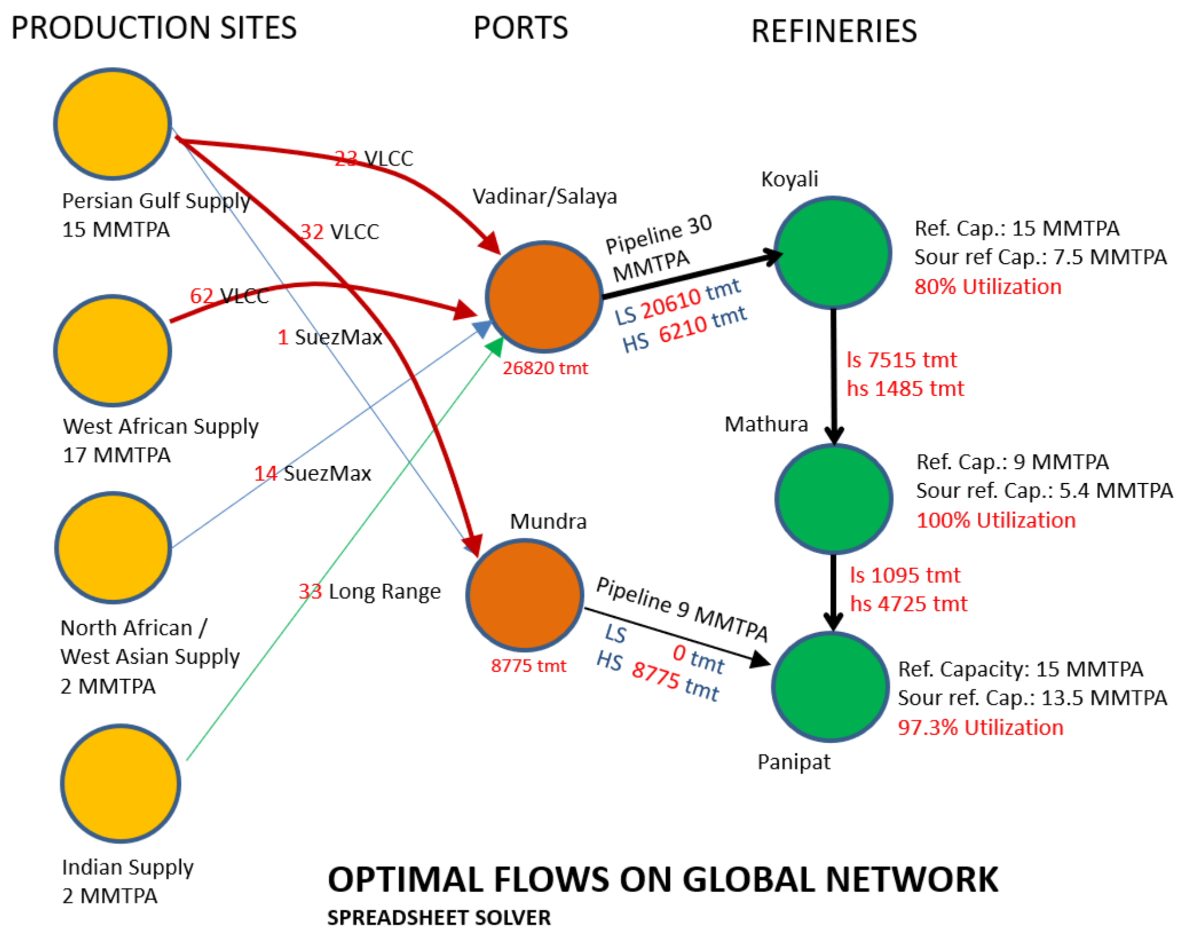

Figure 1 Optimal shipping pattern and pipeline flows.

2. Materials and methods

The classic transshipment problem is known to all students of mathematical programming. An interesting and recent application in the carbonated beverage industry is available in the paper by Agadaga and Akpan (2019). In their case, direct routing turned out to be more economical than using the transshipment nodes. Tagawa et al. (2021) also examined the alternative of direct shipping and transshipment in their paper. Sherali et al. (1999) analyzed fleet management models and algorithms for oil tanker routing and scheduling. Vilhelmsen et al. (2015) worked on tramp ship routing and scheduling.

McCalla (2008) studied Kingston, Jamaica, as an important transshipment hub. Ryuichi et al. (2017) looked at global route choices, with the Suez Canal as the focus. Dávid et al. (2021) wrote about transshipment activities in the Rhine ports. Diz et al. (2017) researched the maritime inventory routing for a Brazilian company.

In recent years, the *Gurobi* optimizer has become quite dominant in the solver segment of the software industry. The idea behind this paper was to re-solve the existing model using the *Gurobi* optimizer. The problem was formulated in an RStudio environment and solved by function calls in the *Gurobi* package. The outputs are as in **Figures 2** and **3**. In **Figure 2**, one can see the initialization and presolve routines. The duality gap reduces to zero in 16 simplex iterations. The optimal solution that I obtained is displayed in **Figure 3**. Though the objective function value was the same as that obtained with the spreadsheet, the flow values were quite different.

```
Gurobi Optimizer version 9.1.1 build v9.1.1rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 29 rows, 40 columns and 120 nonzeros
Model fingerprint: 0x4f3e7fe4
Variable types: 30 continuous, 10 integer (0 binary)
Coefficient statistics:
  Matrix range     [9e-01, 3e+02]
  Objective range  [8e+03, 2e+07]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+03, 3e+04]
Found heuristic solution: objective -0.0000000
Presolve removed 13 rows and 18 columns
Presolve time: 0.00s
Presolved: 16 rows, 22 columns, 60 nonzeros
Variable types: 12 continuous, 10 integer (0 binary)

Root relaxation: objective 8.967480e+08, 11 iterations, 0.00 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 8.9675e+08    0    3   -0.00000 8.9675e+08     -      -    0s
H    0     0                    8.945473e+08 8.9675e+08  0.25%     -    0s
     0     0 8.9625e+08    0    4 8.9455e+08 8.9625e+08  0.19%     -    0s
H    0     0                    8.957527e+08 8.9625e+08  0.06%     -    0s
H    0     0                    8.959328e+08 8.9625e+08  0.04%     -    0s
     0     0 8.9625e+08    0    2 8.9593e+08 8.9625e+08  0.04%     -    0s
     0     0 8.9625e+08    0    4 8.9593e+08 8.9625e+08  0.04%     -    0s
*    0     0               0    8.962480e+08 8.9625e+08  0.00%     -    0s
```

Figure 2 Presolve and starting nodes of the branch and bound.

```
Cutting planes:
  MIR: 4

Explored 1 nodes (16 simplex iterations) in 0.01 seconds
Thread count was 8 (of 8 available processors)

Solution count 5: 8.96248e+08 8.95933e+08 8.95753e+08 ... -0

Optimal solution found (tolerance 1.00e-04)
Best objective 8.962479875000e+08, best bound 8.962479875000e+08, gap 0.0000%
> print(result$objval)
[1] 896247988
> print(result$x)
 [1] 19579.50 14235.75    55.00     0.00    31.00    31.00     0.00     1.00    13.00
[10]     1.00    27.00     6.00 11745.00 14850.00  8865.00   135.00     0.00     0.00
[19]     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
[28]     0.00  7500.00  3600.00   645.00  7500.00  5400.00  1950.00     0.00     0.00
[37]  8865.00     0.00     0.00   135.00
> 
```

Figure 3 Results of the *Gurobi* Optimizer run.

3. Results and discussion

First, I recapitulate the results of the original work. The optimal product mix obtained for the transshipment problem by the spreadsheet solution, under sensitivity analysis, was stable within a large interval of both profit margins. The effects of changes in refinery margin and in crude oil availability on shipping pattern are also available through the solver table. One can see, in **Figure 1**, that the *Koyali* refinery was used 80 %, and that the *Mathura* refinery was used 100 %. The *Panipat* refinery was used 97.3 %. Such results are useful for capacity expansion decisions. It was concluded that the mathematical model can be expanded to include new suppliers, ports, and freight options which have gained prominence in the 21st century, due to various political, financial, technological, or strategic reasons.

The optimal objective value obtained from the *Gurobi* optimizer differed by less than 0.1 % from the objective value obtained from the spreadsheet solver. The product mix of the two refined products was also the same. However, the optimal shipping pattern was different, as can be seen by comparing **Figure 4** below with **Figure 1**. One can see that the utilization of the refineries is different in the two cases. In the second solution, the *Koyali* and *Mathura* refineries were used 100 %, and the Panipat refinery was used only 77.3 %. Suppose that a refinery is down for maintenance or repair. Then, the entire 100 % capacity will not be available for that period. If possible, one would give more feed to a refinery that is functioning at 100 %. The two optimal solutions suggest that this prioritization can be done upstream of the network, at the supply ports itself.

It is a known fact in mathematical programming that, if there are two alternate optimal solutions, then there are an infinite number of alternate optimal solutions in between them. In this case, the integer restriction means I do not have an infinite number of solutions but, nevertheless, I have more than two solutions. However, the solutions obtained by rounding a convex combination solution is not far from the optimal solution. All of them are a fraction of a percentage point over the optimal solution. Only the solution for the midpoint of the two solutions is 0.6 % lower than the optimal solution, as can be seen in **Table 3**. These alternatives are all feasible to the constraints of the problem. Each alternate optimal solution would affect the utilization of the refineries somewhat differently. This fact can be used to the oil company’s advantage for planning purposes. One can determine the right alternative that suits the existing or future refining capacity. This can be done on a dynamic basis throughout the year in order to schedule maintenance at the refineries.

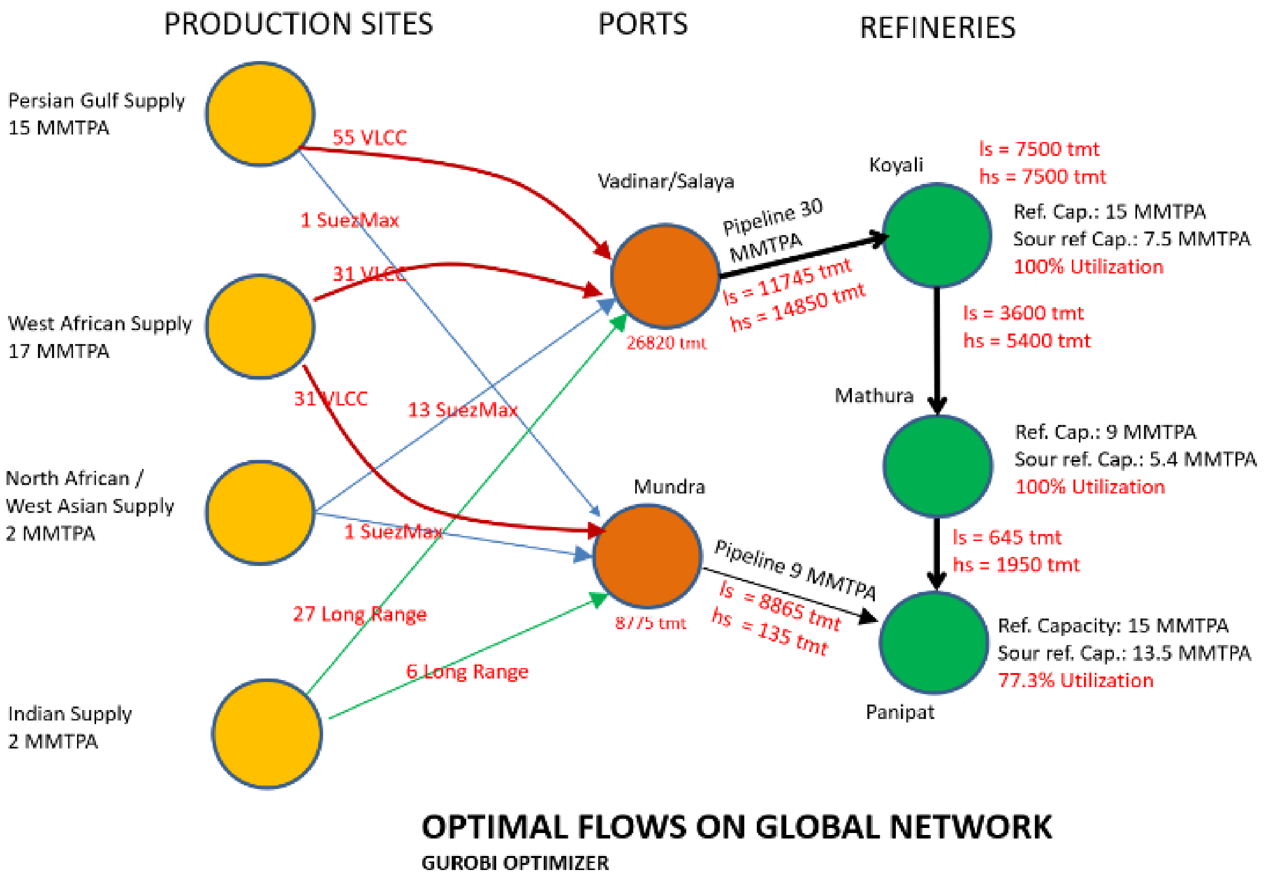

Figure 4 Optimal shipping pattern and pipeline flows using *Gurobi* optimizer.

Table 3 Convex combinations of the alternate optimal solutions and deviation from solution.

α	V1	V2	V3	V4	S1	S2	S3	S4	LR1	LR2	% Deviation from optimality	% Deviation for rounded solution
0.0	55	0	31	31	0	1	13	1	27	6	0.075	0.075
0.1	51.8	3.2	34.1	27.9	0	1	13.1	0.9	27.6	5.4	0.068	0.068
0.2	48.6	6.4	37.2	24.8	0	1	13.2	0.8	28.2	4.8	0.060	0.060
0.3	45.4	9.6	40.3	21.7	0	1	13.3	0.7	28.8	4.2	0.053	0.053
0.4	42.2	12.8	43.4	18.6	0	1	13.4	0.6	29.4	3.6	0.045	0.045
0.5	39	16	46.5	15.5	0	1	13.5	0.5	30	3	0.038	-0.576
0.6	35.8	19.2	49.6	12.4	0	1	13.6	0.4	30.6	2.4	0.030	0.030
0.7	32.6	22.4	52.7	9.3	0	1	13.7	0.3	31.2	1.8	0.023	0.023
0.8	29.4	25.6	55.8	6.2	0	1	13.8	0.2	31.8	1.2	0.015	0.015
0.9	26.2	28.8	58.9	3.1	0	1	13.9	0.1	32.4	0.6	0.008	0.008
1.0	23	32	62	0	0	1	14	0	33	0	0	0.0

Table 4 displays the capacity utilization for the three refineries across the eleven shipping patterns. The alpha parameter in the first column generates a linear combination of the two optimal solutions. For example, one can see with the *Koyali* refinery that the low sulfur refining capacity utilization increases linearly from 50 to 100 %, in increments of 5 %, whereas the high sulfur refining capacity utilization drops to zero, in increments of 5 %. This means that, if the high sulfur refining capacity is not available due to maintenance or expansion work, one may prefer the alternatives near the bottom of the table. For the *Mathura* refinery, the effect is almost linear, and the utilization proportion climbs from 40 to 60 % for low sulfur and drops from 60 to 40 % for high sulfur. For the *Panipat* refinery, too, the effect is almost linear, with the low sulfur capacity falling from 82 % to zero, as I look down the table, and the high sulfur capacity utilization climbing from 18 to 100 %. A manager may use a status report from all three refineries to pick a shipping alternative that is best.

Table 4 Capacity utilization at three refineries for different shipping alternatives.

α	Low sulfur util. at Koyali	High sulfur util. at Koyali	Low sulfur util. at Mathura	High sulfur util. at Mathura	Low sulfur util. at Panipat	High sulfur util. at Panipat
0	50	50	40	60	82	18
0.1	55	45	42	58	74	26
0.2	60	40	44	56	66	34
0.3	65	35	47	53	57	43
0.4	70	30	49	51	49	51
0.5	75	25	51	49	41	59
0.6	75	25	53	47	33	67
0.7	85	15	56	44	25	75
0.8	90	10	58	42	16	84
0.9	95	5	60	40	8	92
1.0	100	0	62	38	0	100

4. Conclusions

The *Gurobi* optimizer gives a solution that is less than 0.1 % higher than the spreadsheet solution, an indication of an alternate optimal solution. Furthermore, a duality gap of zero is proof that it is the global optimal solution. If neither of the two optimal solutions can be implemented, a convex combination, wherein the continuous values are rounded off to integer solutions, may be satisfactory to both the refinery and the transporter. The alternate optimal solutions, thus, have practical advantages that the extreme point solutions may not have. Given the current situation, a weighting scheme, such as those used in Multicriteria Decision Making (MCDM), can be employed to obtain a score and rank for each alternative.

Acknowledgement

I thank the editorial team of the Maritime Technology and Research journal for the opportunity to share this discovery.

References

Agadaga, G. O., & Akpan, N. P. (2019). A transshipment model of Seven-Up Bottling Company, Benin Plant, Nigeria. *American Journal of Operations Research, 9*(3), 129-145. https://doi.org/10.4236/ajor.2019.93008

Dávid, A., Mako, P., Galieriková, A., & Stupalo, V. (2021). Transshipment activities in the Rhine Ports. *Transportation Research Procedia, 53*, 188-195, https://doi.org/10.1016/j.trpro.2021.02.025

Diz, G. S. S., Oliveira, F., & Hamacher, S. (2017). Improving maritime inventory routing: Application to a Brazilian petroleum case. *Maritime Policy & Management, 44*(1), 42-61. https://doi.org/10.1080/03088839.2016.1216622

McCalla, R. J. (2008). Container transshipment at Kingston, Jamaica. *Journal of Transport Geography, 16*(3), 182-190. https://doi.org/10.1016/j.jtrangeo.2007.05.006

Ryuichi, S., Azuma, T., Yoshida, T., Teranishi, H., & Abe, M. (2017). Global route choice and its modelling of dry bulk carriers based on vessel movement database: Focusing on the Suez Canal. *Research in Transportation Business & Management, 25*, 51-65. https://doi.org/10.1016/j.rtbm.2017.08.003

Sherali, H. D., & Al-Yakoob, S., & Hassan, M. M. (1999). Fleet management models and algorithms for an oil-tanker routing and scheduling problem. *IIE Transactions, 31*, 395-406. https://doi.org/10.1023/A:1007502205917

Tagawa, H., Kawasaki, T., & Hanaoka, S. (2021). Conditions influencing the choice between direct shipment and transshipment in maritime shipping network. *Journal of Shipping and Trade, 6*, 4. https://doi.org/10.1186/s41072-021-00085-3

Toppur, B., & Sanyal, A. (2020). A transshipment model for logistics management at Indian Oil Corporation. *RAIRO-Operations Research, EDP Sciences*, *54*(4), 1087-1102. https://doi.org/10.1051/ro/2020005

Vilhelmsen, C., Larsen, J., & Lusby, R. M. (2015). *Tramp ship routing and scheduling: Models, methods and opportunities* (pp. 1-33). DTU Management Engineering.

Appendix

Serial No.	Variable Name	Definition in Mixed Integer Program
1	G_1 and G_2	TMTs of product from 2 types of crude oil
2	V_1 and V_2	VLCC from Persian Gulf to Salaya/Vadinar and Mundra
3	V_3 and V_4	VLCC from West Africa to Salaya/Vadinar and Mundra
4	S_1 and S_2	Suezmax from Persian Gulf to Salaya/Vadinar and Mundra
5	S_3 and S_4	Suezmax from N. Africa/W. Asia to Salaya/Vadinar and Mundra
6	LR_1 and LR_2	LR from Bombay High to Salaya/Vadinar and Mundra
7	LV	TMT of sweet crude oil delivered at Salaya/Vadinar terminal
8	HV	TMT of sour crude oil delivered at Salaya/Vadinar terminal
9	LM	TMT of sweet crude oil delivered at Mundra terminal
10	HM	TMT of sour crude oil delivered at Mundra terminal
11	$LV_{1, r}$	Sweet crude oil from Salaya/Vadinar to Koyali by rail
12	$LV_{2, r}$	Sweet crude oil from Salaya/Vadinar to Mathura by rail
13	$LV_{3, r}$	Sweet crude oil from Salaya/Vadinar to Panipat by rail
14	$HV_{1, r}$	Sour crude oil from Salaya/Vadinar to Koyali by rail
15	$HV_{2, r}$	Sour crude oil from Salaya/Vadinar to Mathura by rail
16	$HV_{3, r}$	Sour crude oil from Salaya/Vadinar to Panipat by rail
17	$LM_{1, r}$	Sweet crude oil from Mundra to Koyali by rail
18	$LM_{2,r}$	Sweet crude oil from Mundra to Mathura by rail
19	$LM_{3,r}$	Sweet crude oil from Mathura to Panipat by rail
20	$HM_{1,r}$	Sour crude oil from Mundra to Koyali by rail
21	$HM_{2,r}$	Sour crude oil from Mundra to Mathura by rail
22	$HM_{3,r}$	Sour crude oil from Mundra to Panipat by rail
23	$LV_{1,p}$	Sweet crude oil from Salaya/Vadinar to Koyali by pipeline
24	$LV_{2,p}$	Sweet crude oil from Salaya/Vadinar to Mathura by pipeline
25	$LV_{3,p}$	Sweet crude oil from Salaya/Vadinar to Panipat by pipeline
26	$HV_{1,p}$	Sour crude oil from Salaya/Vadinar to Koyali by pipeline
27	$HV_{2,p}$	Sour crude oil from Salaya/Vadinar to Mathura by pipeline
28	$HV_{3,p}$	Sour crude oil from Salaya/Vadinar to Panipat by pipeline
29	$LM_{1,p}$	Sweet crude oil from Mundra to Koyali by pipeline

Serial No.	Variable Name	Definition in Mixed Integer Program
30	$LM_{2,p}$	Sweet crude oil from Mundra to Mathura by pipeline
31	$LM_{3,p}$	Sweet crude oil from Mundra to Panipat by pipeline
32	$HM_{1,p}$	Sour crude oil from Mundra to Koyali by pipeline
33	$HM_{2,p}$	Sour crude oil from Mundra to Mathura by pipeline
34	$HM_{3,p}$	Sour crude oil from Mundra to Panipat by pipeline