Truck Load Problem

Saksham Kumar
Roll No.:200102109
IIT, Guwahati
Guwahati, India
k.saksham@iitg.ac.in

Akshaj Padmakar
Roll No.:200108005
IIT, Guwahati
Guwahati, India
p.akshaj@iitg.ac.in

Samarth Umapati Hegde
Roll No.:200102081
IIT, Guwahati
Guwahati, India
h.samarth@iitg.ac.in

***Abstract*—The truck load problem solved in this work involves maximizing the minimum load that can be delivered to a destination in case of a crane failure and simultaneously minimizing the maximum cost to deliver the load to a destination. In this work, a non-linear mathematical model is developed and multi-objective optimization is performed using the epsilon-constraint method.**

I. Introduction

In this problem, there are trucks at a port, which are to be loaded with goods and delivered to certain destinations. Each truck has different parameters like the load it can carry i.e **capacity**, the space it occupies in the parking lot i.e **length**, and the distance it covers per unit cost i.e **mileage**. Each destination is characterized by its **distance** from the port.

The parking lot is divided into rows, each having the same number of slots, where each slot is of unit length. Some slots are unavailable due to other establishments in the parking lot, thus cannot be used for parking. For the truck to be loaded it has to be parked in consecutive slots in the same row such that it doesn't occupy any unavailable slots. One crane is responsible for loading all trucks parked in a particular row.

Some trucks are grouped together irrespective of their positions in the parking lot, to deliver to a specific destination. Each truck can be assigned to at most one group.

Since cranes used for loading in the port are old and prone to malfunction, there might be a case where a particular crane can fail leading to all the trucks parked in the corresponding row not being loaded, resulting in a reduced load that can be delivered to destinations that the trucks of that row were assigned to.

Sure load is defined as the minimum load that can be delivered to any destination in case any one of the employed cranes malfunctions.

Total cost for delivering to a destination is given by the summation of the cost incurred by each truck delivering to that destination.

The objective of the problem is to maximize the sure load and simultaneously minimize the maximum total cost to a certain destination by optimally assigning parking space and destination to a truck in the parking lot. Some trucks might be left unassigned.

Mathematical model for the problem is formulated using Mixed Integer Non-Linear Programming.

The solver employed is **GAMS** software, **C++** is used to pre-process the data and filter solutions in the first Pareto front and **MATLAB** is used to plot the first Pareto front having the non-dominated solutions.

II. Problem Description

The truck load problem can be defined as follows.
Given:

- number of **rows** in the parking lot
- number of columns(**slots**) in each row
- number of **unavailable slots**
  - Position(index) of each unavailable slot in the parking lot
- number of trucks
  - **Capacity** of each truck
  - **Length** of each truck
  - **Mileage** of each truck
- number of destinations
  - **Distance** of each destination

For convenience in implementation, the data is processed before being fed into the formulation. The given data of available and unavailable slots is converted to **blocks** where each block is formed by clubbing adjacent available slots in a row. The number of blocks in a row cannot be more than the number of columns. Maintaining the dimension of data through pre-processing will result in some 0-sized blocks in each row. More than one trucks can be parked in a block provided their total length does not exceed the block size.

A feasible solution should also additionally satisfy the following restrictions:

1) One truck can be parked in at most one block
2) One truck can be assigned to at most one destination
3) A truck will be assigned to a destination only if it is parked in a block
4) Sum of lengths of the trucks parked in a block should not exceed the size of that block.

III. Mathematical Model

The proposed formulation requires the following indices, sets, parameters, and variables.

A. *Indices and Sets*

- $\mathbf{i} \in \mathbf{I}$: rows in parking lot
- $\mathbf{j} \in \mathbf{J}$: columns in each row
- $\mathbf{k} \in \mathbf{K}$: truck
- $\mathbf{l} \in \mathbf{L}$: destinations

B. *Parameters*

- $\mathbf{C_k}$: capacity of k^{th} truck
- $\mathbf{len_k}$: length of k^{th} truck
- $\mathbf{mil_k}$: mileage of k^{th} truck
- $\mathbf{dist_l}$: distance of l^{th} destination
- $\mathbf{z_{(i,j)}}$: size of $(i,j)^{th}$ block in the parking lot

C. *Decision Variables*

All decision variables are binary.

$$\mathbf{a(i,j,k)} = \begin{cases} \mathbf{1} & \text{if } k^{th} \text{ truck is parked at } (i,j)^{th} \text{ block in the parking lot} \\ \mathbf{0} & \text{Otherwise} \end{cases}$$

$$\mathbf{b(k,l)} = \begin{cases} \mathbf{1} & \text{if } k^{th} \text{ truck is assigned to } l^{th} \text{ destination} \\ \mathbf{0} & \text{Otherwise} \end{cases}$$

D. *Constraints*

As stated $\mathbf{a(i,j,k)}$ is a binary variable which assumes $\mathbf{1}$ if the $\mathbf{k^{th}}$ truck with index is parked to the block at index $\mathbf{(i,j)^{th}}$ block, otherwise $\mathbf{0}$. A feasible solution should satisfy the constraint that each truck should be parked in at most one block only, which can be expressed as :

$$\boxed{\sum_i \sum_j a(i,j,k) \leq 1, \ \forall k} \tag{1}$$

As stated $\mathbf{b(k,l)}$ is a binary variable which assumes $\mathbf{1}$ if the $\mathbf{k^{th}}$ truck is assigned to $\mathbf{l^{th}}$ destination, otherwise $\mathbf{0}$. A feasible solution should satisfy the constraint that each truck should be assigned to at most one destination only, which can be expressed as :

$$\boxed{\sum_l b(k,l) \leq 1, \ \forall k} \tag{2}$$

A feasible solution should also satisfy the constraint that a truck should be assigned to a destination only if it is parked in a block in the parking lot, which can be expressed as :

$$\boxed{\sum_l b(k,l) = \sum_i \sum_j a(i,j,k), \ \forall k} \tag{3}$$

As stated $\mathbf{len_k}$ is the length of $\mathbf{k^{th}}$ truck and $\mathbf{z(i,j)}$ is the size of the $\mathbf{j^{th}}$ block in the $\mathbf{i^{th}}$ row. A feasible solution should satisfy the constraint that the sum of all the lengths of all trucks parked in a particular block shouldn't exceed the length of the block, which can be expressed as :

$$\boxed{\sum_k len_k \cdot a(i,j,k) \leq z(i,j), \ \forall i, \ j} \tag{4}$$

It can be observed that the number of constraints can be reduced by 1 as one of the constraints (1) or (2) can be omitted as one of them along with constraint (3) will imply the other.

E. *Objective function*

- **Maximizing** the sure load, where sure load is given as :

$$\min_{i,l} \sum_k c_k \cdot b(k,l) - \sum_k c_k \cdot b(k,l) \cdot \sum_j a(i,j,k)$$

- **Minimizing** the maximum total cost to any destination which is given as :

$$\max_l \sum_k (dist_l / mil_k) \cdot b(k,l)$$

Handling MINI-MAX and MAXI-MIN objective functions: As seen one of the objective functions is MINI-MAX and the other is MAXI-MIN, we can remodel them for implementational convenience as as follows :
Let,

$$g(i,l) = \sum_k c_k \cdot b(k,l) - \sum_k c_k \cdot b(k,l) \cdot \sum_j a(i,j,k)$$

Here we can interpret $g(i,l)$ as, load delivered to the l^{th} destination on failure of the crane corresponding to i^{th} row.
A constraint is added to the problem as follows:

$$g(i,l) \geq sl$$

Here sl is the sure load.
Thus for a feasible solution, sl will be the minimum of g(i,l).
Therefore the objective becomes **maximizing** sl.

Similarly for the second objective function :
Let,

$$Cost(l) = \sum_k (dist_l / mil_k) \cdot b(k,l)$$

Here $Cost(l)$ can be interpreted as,the total cost of delivering to l_{th} destination.
A constraint is added to the problem as follows:

$$Cost(l) \leq tc$$

Here tc is the maximum cost of delivering to any destination. Thus for a feasible solution, tc will be the maximum of $Cost(l)$. Therefore the objective becomes **minimizing** tc.

F. Solving Multi-Objective Problem

ϵ-constraint method is employed for simultaneously optimizing the two objective functions. One of the objective functions is restricted, resulting in an additional constraint under which the other objective function is optimized.
Thus, being a minimization objective, tc is restricted as,

$$tc \leq \epsilon$$

and objective sl is maximized under the above constraint.
Different solutions are obtained by varying ϵ from maximum value of tc to minimum value of tc which can be obtained by maximizing and minimizing tc without any additional constraints.
The size of the steps in which ϵ varies can be adjusted. Smaller the step size, more the number of solutions generated. Thus, varying ϵ in smaller steps might lead to better solutions.
The solutions generated lie on different pareto-fronts, where each pareto front consists of non-dominated solutions. Solutions lying on the first pareto front dominate those lying on other fronts. Out of those, solutions with more crowding distance can be filtered out.

IV. Implementation in GAMS

```
$onInline
sets
i /1,2,3/
j /1,2,3,4,5/
k /1,2,3,4,5,6,7/
l /1,2/;

/*Here,
    i -> number of rows
    j -> number of columns
    l -> number of destination
    k -> number of trucks
*/

Parameter
c(k)
/   1 10
    2 5
    3 16
    4 8
    5 7
    6 3
    7 9/
/*c(k) is the capcity of the k-th truck.*/

len(k)
/   1 3
    2 1
    3 4
    4 2
    5 2
    6 1
    7 3/
/*len(k) is the length of the k-th truck.*/

mil(k)
/   1 5
    2 10
    3 8
    4 20
    5 15
    6 5
    7 10/
/*mil(k) is the mileage of the k-th truck.*/

dist(l)
/   1 50
    2 100/;
/*dist(l) is the distance l-th truck.*/

Scalar
MaxCost /84.16667/;

Table z(i,j)
    1 2 3 4 5
1   4 0 0 0 0
2   1 2 0 0 0
3   1 2 0 0 0;
/*z(i, j) -> size of the (i,j)-th block in the parking lot.*/

Variables
a(i,j,k)
b(k,l)
sl
tc;
/*Here,
        a(i, j, k) = kth truck is parked at the (i,j)th block in the parking lot.

        b(k, l) = k-th truck is assigned to l-th truck destination.

        sl = sure load
        tc = maximum cost of delivering to any destination.
*/


Integer Variables
a(i,j,k)
b(k,l);


Equations
load(i,l)
cost(l)
block(k)
destination(k)
block_and_destination(k)
block_size(i,j);


load(i,l).. sum(k,c(k)*b(k,l))-sum(k,c(k)*b(k,l)*(sum(j,a(i,j,k))))=g=sl;

cost(l).. sum(k,(dist(l)/mil(k))*b(k,l))=l=tc;

block(k).. sum((i,j),a(i,j,k))=l=1;

destination(k).. sum(l,b(k,l))=l=1;

block_and_destination(k).. sum((i,j),a(i,j,k))-sum(l,b(k,l))=e=0;

block_size(i,j).. sum((k),len(k)*a(i,j,k))=l=z(i,j);


a.lo(i,j,k)=0;
b.lo(k,l)=0;

a.up(i,j,k)=1;
b.up(k,l)=1;

a.l(i,j,k)=0;
b.l(k,l)=0;
```

```
sl.l=0;
tc.l=0;


Model multi_obj / all /;

Set counter / c1*c41 /;

Scalar E;

Parameter report(counter,*), ranges(*);


/* finding maximum(worst) value of TC by maximizing
     SL without any additional constraints*/
solve multi_obj using minlp maximizing sl;
ranges('SLmax') = sl.l;
ranges('TCmax') = tc.l;


/* finding minimum(best) value of TC by minimizing
     TC without any additional constraints*/
solve multi_obj using minlp minimizing tc;
ranges('TCmin') = tc.l;
ranges('SLmin') = sl.l;



/*
    solving multi-objective by restricting TC
    epsilon value(E) is varied from TCmin to TCmax
*/

 loop(counter,
   E = (ranges('TCmax') - ranges('TCmin'))*(ord(
     counter)-1)/(card(counter) - 1)+ranges('TCmin');
   tc.up = E;
   solve multi_obj using minlp maximizing sl;
   report(counter,'SL') = sl.l;
   report(counter,'TC') = tc.l;
   report(counter,'E')   = E;
);

display ranges;
display report;
```

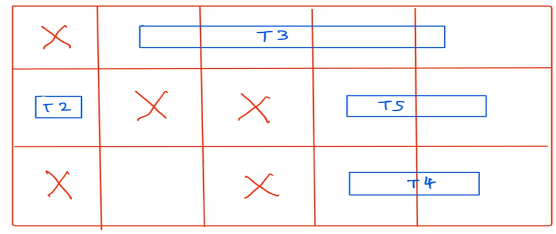
V. Solutions to Sample Data Sets

Example 1 :

- Number of rows, $i = 3$
- Number of columns, $j = 5$
- Number of Unavailable slots, $u = 3$
- Number of Destinations, $l = 2$
- Number of Trucks, $k = 7$
- Coordinates of Unavailable slots, $(1, 1)$, $(2, 2)$, $(2, 3)$, $(3, 1)$, $(3, 3)$.
- Capacity of trucks, $10, 5, 16, 8, 7, 3, 9$
- Length of trucks, $3, 1, 4, 2, 2, 1, 3$
- Mileage of trucks, $5, 10, 8, 20, 15, 5, 10$
- Distance of destinations, $50, 100$

Now the best solutions through GAMS code found are shown below :

Solution1 SL = 5 TC = 11.67



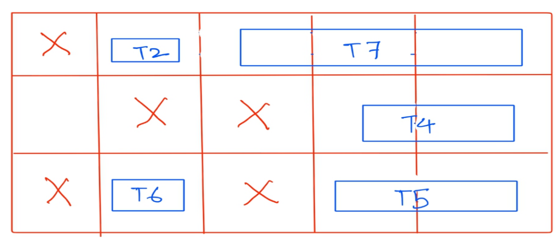
destination

1 → T2, T3, cost1 = 11.25

2 → T4, T5, cost2 = 11.67 → TC

row removed	load to each destination
1	1 → 5 → SL
	2 → 15
2	1 → 16
	2 → 8
3	1 → 21
	2 → 7

Solution 2 : SL = 7, TC = 17.500



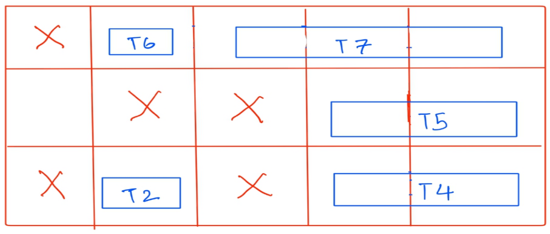
destination

1 → T2, T4, T6 → Cost 1 = 17.5 → Tc

2 → T5, T7 → Cost 2 = 16.66

row removed	load to each destination
1	1 → 11
	2 → 7 → SL
2	1 → 8
	2 → 16
3	1 → 13
	2 → 9

Solution 2, SL = 8   TC = 18.33



destination

1 → T2, T5, T6 , Cost 1 = 18.33 → T

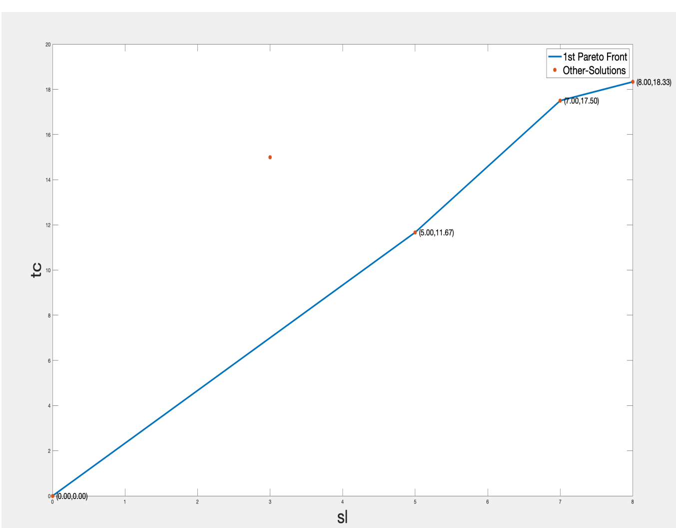
2 → T4, T7 , Cost 2 = 15

row removed	load to each destination
1	1 → 12
	2 → 8
2	1 → 8 → SL
	2 → 17
3	1 → 10
	2 → 9

The non-dominated solutions obtained on the 1^{st} Pareto Front are as follows :

1^{st} Pareto Front Solutions	
SL	TC
0	0
5	11.667
7	17.5
8	18.333

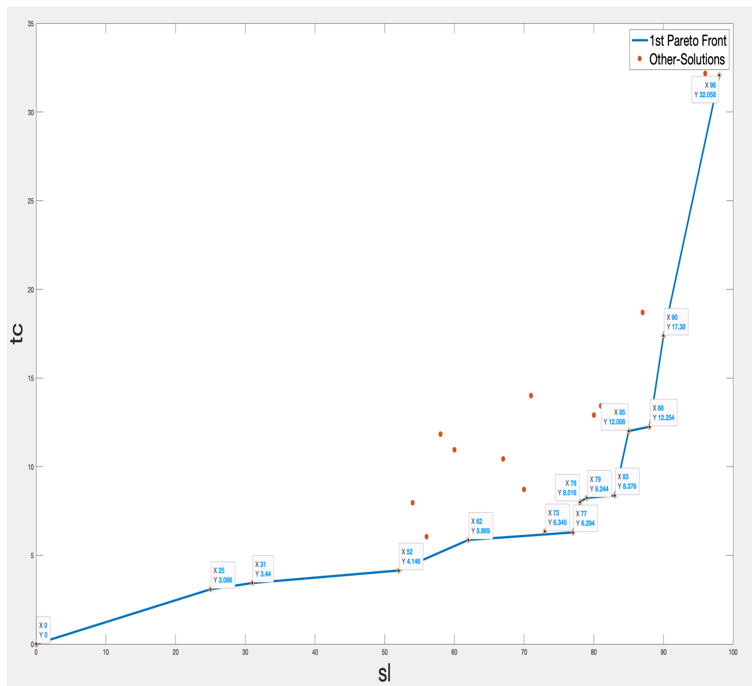
The plot for 1st-Pareto Front and Other Solutions for the above Example is shown below :



Example 2 :

- Number of rows, $i = 5$
- Number of columns, $j = 9$
- Number of Unavailable slots, $u = 1$
- Number of Destinations, $l = 3$
- Number of Trucks, $k = 12$
- Coordinates of Unavailable slots, $(1, 2)$.
- Capacity of trucks, $78, 13, 31, 52, 58, 25, 70, 31, 59, 36, 29, 54$.
- Length of trucks, $1, 4, 4, 2, 4, 1, 4, 3, 3, 1, 4, 3$.
- Mileage of trucks, $12, 34, 92, 36, 23, 86, 97, 36, 1, 16, 12, 42$.
- Distance of destinations, $97, 89, 31$

The plot for 1^{st}-Pareto Front and Other Solutions for the above Example is shown below :



The non-dominated solutions obtained on the 1^{st} Pareto Front are as follows :

1^{st} Pareto Front Solutions	
SL	TC
0	0
25	3.086
31	3.44
52	4.146
62	5.869
77	6.294
78	8.016
79	8.244
83	8.376
85	12.008
88	12.254
90	17.38
98	32.058

Optimization Problem size for a given Input Data size

Statistical Table for No. of Decision Variables					
No. of Rows (i)	No. of Columns (j)	No. of Blocks (i*j)	No. of Trucks (k)	No. of Destinations (l)	Decision Variables (i*j*k + k*l)
3	5	15	7	2	119
5	9	45	12	3	576
6	11	66	15	5	1065
8	10	80	18	6	1548
11	14	154	21	7	3381
15	15	225	25	10	5875

Statistical Table for No. of Constraints							
Rows (i)	Columns (j)	Trucks (k)	Destinations (l)	Constraints(I) (k)	Constraints(II) (k)	Constraints(III) (k)	Constraints(IV) (i*j)
3	5	7	2	7	7	7	15
5	9	12	3	12	12	12	45
6	11	15	5	15	15	15	66
8	10	18	6	18	18	18	80
11	14	21	7	21	21	21	154
15	15	25	10	25	25	25	225

VI. Conclusion

The multi-objective truck load problem is solved using non-linear mathematical model. The MINI-MAX and MAXI-MIN objective functions were modified to constraints for implementational convenience. Multiple objective functions were optimized simultaneously using the ϵ constraint technique. The non-dominated solutions lying on the 1st Pareto Front are chosen. Further, solutions with higher crowding distance can be filtered out to give a diverse set of non-dominated solutions.

References

[1] Google Hashcode 2015 Qualification round.
[2] Pareto-Front GAMS.
[3] ϵ constrained method.
[4] MINI-MAX and MAXI-MIN constraints.

Appendix

GitHub Repository.