**Parabola Volume 60, Issue 3 (2024)**

# Industrial application of linear optimisation in possible facility and pharmacy locations using JuMP

**Revanth Sainath Killamsetty**[1]

## 1 Introduction

Linear optimisation is a set of algorithms and programs which can be used where the objective function and constraints appear as linear functions of a decision variable [3]. The constraints are usually in the form of linear inequalities. The objective function is a function which calculates and outputs the minimum point or maximum point of a formula using decision variables and iteration. The optimisation program outputs the values for decision variables which can be computed to obtain the result for the objective function. The decision variables can be in the form of vectors or matrices to allow for simpler computation and for easier assignment of decision variables and constraints in the model. Matrices and vectors are fundamental concepts in linear algebra that allow for solving complex problems using simpler computations and methods [6].

In this project, linear optimisation is applied to a specific scenario of a general problem of setting up facility locations to optimise the service towards pharmacies. The programming environment used is `JuMP` [7], a specialised modelling environment embedded in `Julia` [5] which supports complex and simple mathematical optimisation through the use of a variety of solvers. The linear optimisation model used is `HiGHS` [4], a specialised software generally used for solving mixed integer linear programming. It uses a simplex algorithm which is a type of algorithm used to solve linear programming models [2]. This project uses `HiGHS` through `JuMP`.

The scenario used in the project is a modified version of the facility-location problem with more realistic information and decision-making to make it useful in industrial day-to-day problems. This problem will be involved in efficient drug allocation from facilities to pharmacies. For this scenario, each facility has a maximum of $4$ trucks, to limit purchasing and fuel costs by the facilities. Each truck is dedicated to only one pharmacy. Therefore, the maximum number of pharmacies to which one facility can service is set to be $4$. The cost required to build a facility is $\$200$. The cost per kilometre is $\$2$. The amount of drugs necessary to meet the pharmacy's quota is not considered to limit the problem, which allows it to be feasible by linear programming models.

For 2022, the year of the scenario, the average fuel cost is $\$0.5$/km [1] and the average truck driver's wages and benefits are $\$0.6$/km [8]. Adding all other operational costs and insurance costs, the average cost per kilometre of drug transport using trucks is $\$2$/km which is the value listed above. The cost to build a drug-producing facility

[1] Revanth Sainath Killamsetty is a high school student at UIA International School of Tokyo.

is assumed to be $\$200$ for developing countries with governments ready to pay huge amounts in subsidies to the business. Using strategic models, it is possible to reduce the average cost of building an actively-functioning drug-producing facility to $\$200$ from $\$500$. Therefore, $\$200$ to build a facility is the value used in the scenario. To limit the costs for the facility, the number of trucks for each facility is set at $4$ trucks per facility. This number is agreed to be realistic and feasible for a facility to run.

The number of pharmacies for this scenario is $15$ and the number of possible facility locations is $10$.

# 2 Methodology

The decision variables, constants, constraints and objective function used in this project are described below.

## Decision variables

| | |
|---|---|
| $x_j$ | a binary variable indicating whether facility $j$ is built at the location |
| $r_{i,j}$ | a binary variable indicating whether the route between facility $i$ and pharmacy $j$ is in commission |

## Constants

| | |
|---|---|
| $P$ | the number of pharmacies, for this scenario set at $15$ |
| $F$ | the number of facility locations, for this scenario set at $10$ |
| $c_j$ | the cost of building facility $j$, for this scenario set at $\$200$ |
| $d_{i,j}$ | the euclidean distance from pharmacy $i$ to facility $j$ calculated by the program |
| $K$ | cost per kilometre for a truck using the delivery route between a pharmacy and a facility, set at $\$2$ |

## Constraints

For all $i = 0, 1, \ldots, P$ and $j = 0, 1, \ldots, F$,

$$r_{i,j},\ x_j \in \{0, 1\} \tag{1}$$

$$\sum_j r_{i,j} = 1 \tag{2}$$

$$\sum_i r_{i,j} \leq 4 \tag{3}$$

$$r_{i,j} \leq x_j \tag{4}$$

Constraint 1 ensures that the two decision variables are binary. Constraints 2 and 3 ensure that each pharmacy is served by exactly one facility and that each facility can only serve at most $4$ pharmacies. Constraint 4 ensures that unopened facilities are

unable to deliver drugs to pharmacies. Constraints 2 and 3 below follow from the definitions of the constants.

$$K = 2 \tag{5}$$

$$c_j = 200 \tag{6}$$

## Objective function

To minimise the cost of transport and facility construction:

$$\min_{x,r} \sum_{i,j} r_{i,j} d_{i,j} K + \sum_{j} c_j x_j \tag{7}$$

## Code

The following libraries are imported to make the optimisation program easier to understand through visualization. The `Random` library is imported to show and prove that the program can be used in any similar situation which stays within the bounds of the program's capability.

```
import HiGHS
import LinearAlgebra
import Plots
import Random
```

The following code creates random locations for $15$ pharmacies and $10$ facilities. When being used in industrial applications, the variables `Xp`, `Yp`, `Xf` and `Yf` are initialised as vector elements giving the locations of the pharmacies and facilities.

```
p = 15
f = 10
Xp = rand(p)
Yp = rand(p)
Xf = rand(f)
Yf = rand(f)
```

Tables 1 and 2 show the values for $X$-coordinates and $Y$-coordinates generated for the pharmacies and possible facility locations.

```
c = 200*ones(f)
```

This creates a vector element consisting of the fixed costs of building a facility which is $\$200$. If the fixed cost is edited, then the coefficient can be edited accordingly.

The following code creates a matrix of distances and costs between the pharmacies and facilities. Without allowing the distance matrix to be multiplied by $100$, all facilities and pharmacies are placed in a plot of area of $1\,\text{km}^2$. As that is unrealistic, all values in the matrix are multiplied by $100$ so that the total area becomes $10000\,\text{km}^2$. This makes the problem more realistic to run the optimiser model on.

| $X$-coordinates | $Y$-coordinates |
|---|---|
| 0.2922960176110069 | 0.405209315376299 |
| 0.6052943890943097 | 0.08191215304860966 |
| 0.8058768636759748 | 0.40403350925586234 |
| 0.33707903568926323 | 0.8636981777573618 |
| 0.8839190890555322 | 0.07419042843469459 |
| 0.3382686114675154 | 0.6843657332659625 |
| 0.15692628033743927 | 0.9781799525537213 |
| 0.8630328134335518 | 0.830529723728726 |
| 0.05686224265701778 | 0.581060872404951 |
| 0.6006969036090088 | 0.827238171928368 |
| 0.3592286769634545 | 0.267247769390898 |
| 0.7307392890710626 | 0.965014161627441 |
| 0.35764687916008964 | 0.3379644781871336 |
| 0.9781803496143763 | 0.03426941081638468 |
| 0.9369931967277386 | 0.9644348930927312 |

Table 1: Coordinates of all pharmacies in this scenario

| $X$-coordinates | $Y$-coordinates |
|---|---|
| 0.9255290649715613 | 0.27155664771910215 |
| 0.9758290625121906 | 0.04404987116814785 |
| 0.18638492801223938 | 0.34025023585865743 |
| 0.6790462997529064 | 0.5840349885373178 |
| 0.5790815801496081 | 0.47793341859286453 |
| 0.8001111336973329 | 0.7263732881010715 |
| 0.7449953696032204 | 0.13273227903391716 |
| 0.817997338018164 | 0.3317792741710669 |
| 0.885383343930299 | 0.29128625396418617 |
| 0.021525648546421694 | 0.9799452303080997 |

Table 2: Possible coordinates of all facilities in this scenario

```
d = zeros(p,f)
for i in 1:p
    for j in 1:f
        d[i,j] = 100*LinearAlgebra.norm([Xp[i]-Xf[j],
        Yp[i]-Yf[j]],2)
    end
end
k = 2*d
```

The following code makes use of the Plots library to help the user visualise the locations of the pharmacies and facilities. The possible locations are shown in Figure 1

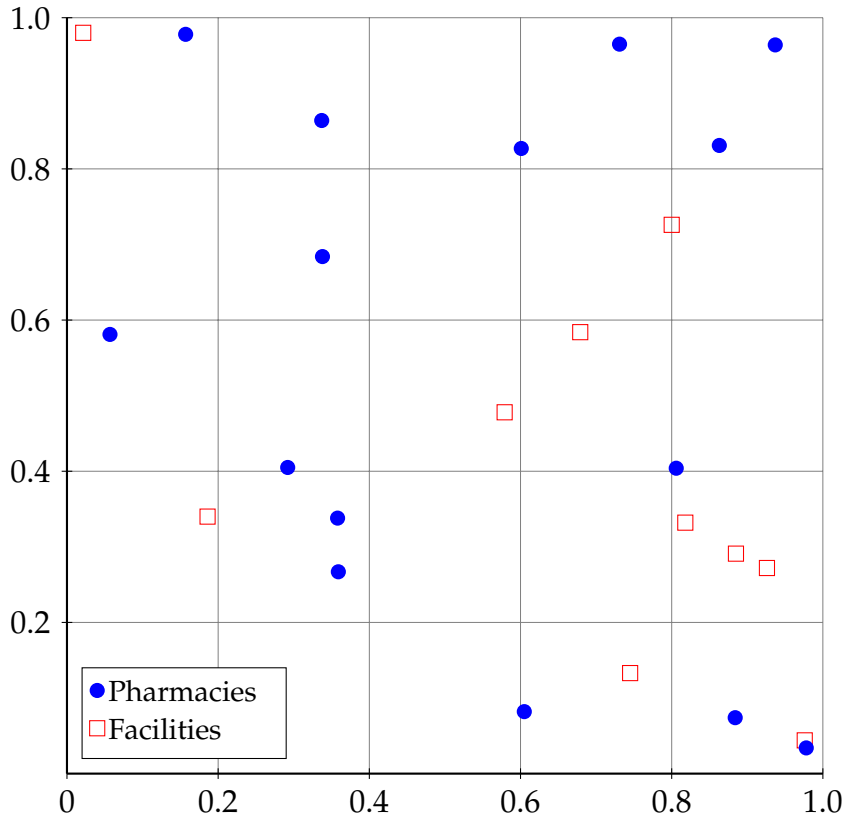


Figure 1: Map of pharmacy locations and facility locations generated

in this specific use of the program.

```
Plots.scatter(
    Xp,
    Yp,
    label = "Pharmacies",
    markershape = :circle,
    markercolor = :blue,
)
Plots.scatter!(
    Xf,
    Yf,
    label = "Facility",
    markershape = :square,
    markercolor = :white,
    markersize = 6,
    markerstrokecolor = :red,
    markerstrokewidth = 2,
)
```

The following code creates a model which uses the `HiGHS` solver. It assigns the decision variables to the solver. It uses vectors and matrices to simplify the problem and the computation required to solve it.

```
model = Model(HiGHS.Optimizer)
@variable(model, x[1:f], Bin)
@variable(model, r[1:p, 1:f], Bin)
```

The following code assigns constraints to the optimiser model, which have not been specified by the previous lines of code.

```
@constraint(model, assignment[i in 1:p],
sum(r[i,j] for j in 1:f)==1);
@constraint(model, service[j in 1:f],
sum(r[i,j] for i in 1:p)<=4);
@constraint(model, open[i in 1:p, j in 1:f], r[i,j]<=x[j]);
```

The following code assigns the objective function to the optimiser model. Then, the model runs its optimiser and provides a solution summary. The solution summary would also include the objective value which is discussed in the results.

```
@objective(model, Min, sum(k .* r)+c'x);
optimise!(model);
solution_summary(model);
```

The following code allows us to visualise the solution by looking at the pharmacies, open and closed facilities.

```
p = Plots.scatter(
    Xp,
    Yp,
    markershape = :circle,
    markercolor = :blue,
    label = nothing,
)
r_ = value.(r) .> 1 - 1e-5
x_ = value.(x) .> 1 - 1e-5

mc = [(x_[j] ? :red : :white) for j in 1:f]
Plots.scatter!(
    Xf,
    Yf,
    markershape = :square,
    markercolor = mc,
    markersize = 6,
```

```
    markerstrokecolor = :red,
    markerstrokewidth = 2,
    label = nothing,
)
```

To view the delivery routes, the following code can be run:

```
for i in 1:p
    for j in 1:f
        if r_[i, j] == 1
            Plots.plot!(
                [Xp[i], Xf[j]],
                [Yp[i], Yf[j]],
                color = :black,
                label = nothing,
            )
        end
    end
end
```

# 3 Results

In this scenario, the final cost for establishing the facility-pharmacy network as shown is the objective value of the optimisation model, namely $\$1472.07703756$. This shows that this will be the minimum cost for establishing a network which pertains to the constraints assigned.

| $X$-coordinates | $Y$-coordinates |
|---|---|
| 0.18638492801223938 | 0.34025023585865743 |
| 0.8001111336973329 | 0.7263732881010715 |
| 0.7449953696032204 | 0.13273227903391716 |
| 0.021525648546421694 | 0.9799452303080997 |

Table 3: Coordinates of all open facilities in this scenario

Table 3 shows the coordinates of the facilities outputted by the optimiser model to be constructed and opened. Figure 2 acts as a visualisation for this table with the results previously obtained.

Figure 2 can be obtained as a result from the code in the Methodology which uses the `Plots` library. This gives us a visualization of which facilities must be opened and which facilities must remain closed. It also gives us a general idea of how it relates with the location of pharmacies.

Figure 3 shows which facilities are assigned to which pharmacy. It is the output from the last block of code in the methodology.

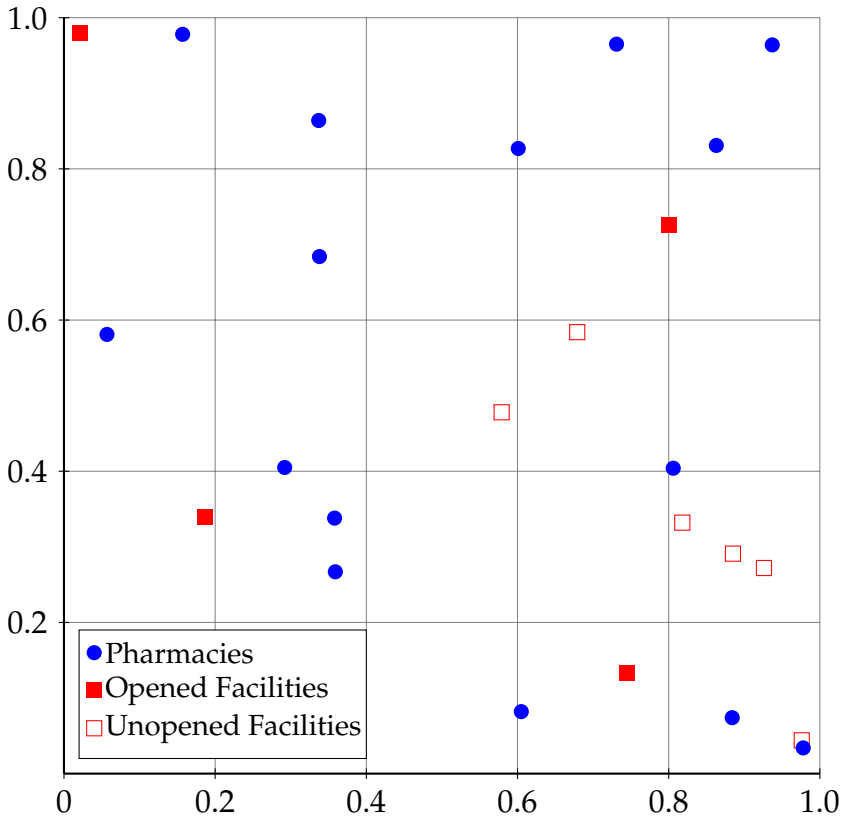


Figure 2: Map of pharmacy locations with open and closed facility locations

# 4 Discussion

The outputs of the program show that it is best for only four out of the ten facilities to be opened. This shows that if more facilities were opened, then unnecessary costs and redundancy would result. If fewer facilities were opened, then not enough pharmacies would be served.

Table 3 shows the coordinates of the facilities which were chosen to be opened.

Based on Figure 3, we understand that the facilities which were relatively closer to at least three pharmacies were chosen to be opened to reduce fuel costs. As stated in the methodology, the fixed cost of building and setting up a facility was assigned at $\$200$. This shows that the total fixed cost for building four factories would be $\$800$.

Additionally, the program outputs that it is best for two facilities to be of service to four pharmacies each, and for the other two facilities to be of service to three pharmacies each. If the limit for each facility's number of pharmacies wasn't set at $4$, then this output could have some difference. However, based on the assigned constraints and objective function, this is shown to be the best choice. While observing Figure 3, it can be noticed that each facility is only paired with pharmacies which are observably closer to them. This is done in order to minimise fuel costs. As fuel cost is assigned to

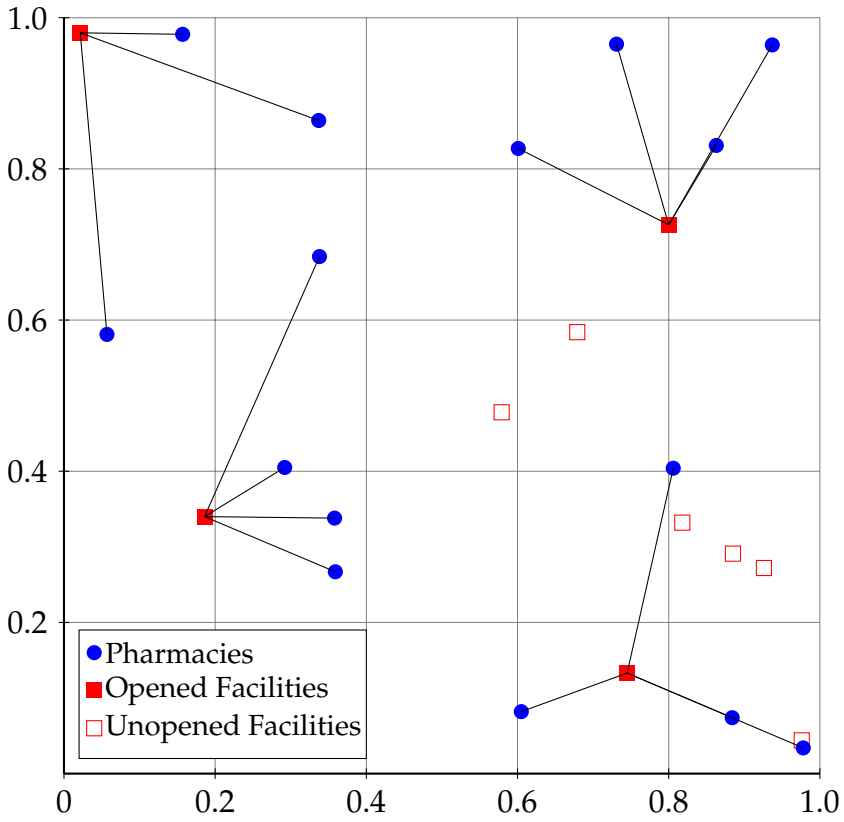


Figure 3: Map with connections between facilities and their assigned pharmacies

be $2/km, the open facility-pharmacy routes let the total fuel cost to be calculated to be approximately $672.08.

Therefore, the total minimum cost outputted by the program is $1472.08. This minimum cost can be obtained by following the solution suggested by the model. Figure 2 and Table 3 show the facilities which need to be opened. The firm using the program can use this information when allocating the funds relevant to opening facilities. Figure 3 shows which routes need to be used. The firm using the program can therefore also use this information when allocating the funds relevant to fuel costs.

This can be applied to any firms in the pharmaceutical industry making use of similar optimisation models. As libraries are used to visualise the solution and make it more comprehensible, there will be little to no error in implementing the solution to minimise their cost. When costs are minimised using the optimiser model, the firm using the program can use their revenue more efficiently as they will have more profit.

# 5 Conclusion

The following results show how the optimiser model can be used to obtain necessary information such as minimum cost and details about which facilities can be opened. Additionally, packages can be used to visualise the results in a way that is easily comprehensible such as the connections between facilities and pharmacies.

This program can be used in a wide-scale industrial approach to minimise costs and maximise profits. With specific parameters, it is possible to obtain similar visualizations to Figure 3 for business expansion proposals and other scenarios where it may be necessary.

However, this specific scenario holds several limitations to the program. The prices for goods vary from country to country and this scenario only accounts for a location which has relatively lower prices for services. Therefore, when using the program in other real-life situations, the parameters must be changed accordingly. Additionally, this program does not account for facilities of different sizes and different number of trucks. This allows for the program to stay simple to be solved using linear optimisation. Another important limitation of this program is that the distances are calculated aerially and not based on roads present in the area. This can be another area where more complex programs can improve by using this optimisation model.

Therefore, this program can only be used to solve simpler delivery optimisation problems. When considering more complex problems related to delivery optimisation, more complex programs must be used. However, this program can be used as a base for more complex programs related to delivery optimisation which can include delivery scheduling, comprehensive cost minimization, and other related problems which can take into account more factors.

# Acknowledgements

This paper was written under the remote guidance of Dr. Benoît Legat through the Cambridge Future Scholar's Program hosted by the Cambridge Centre for International Research. This author would also like to thank teaching assistants Mr. Sibren Lagauw and Mr. Lukas Vanpoucke for their advice when using the solver.

# References

[1] American Transport Research Institute, *Operational Costs of Trucks*, *https://truckingresearch.org/*, 2023.

[2] G. Christofaro and L. Nichols, Explanation of Simplex Method, *IMSE Concept Library*, Iowa State University Publications, 2015.

[3] D. Hsiao, J. Ou and H. S. Sue, Linear Optimization, *Engineering Libretexts*, last accessed on 2024-12-25.

[4] Q. Huangfu and J.A.J. Hall, Parallelizing the dual revised simplex method, *Mathematical Programming Computation* **10** (2018), 119–142.

[5] J. Bezanson, A. Edelman, S. Karpinski and V.B. Shah, Julia: A Fresh Approach to Numerical Computing, *https://julialang.org*, 2017.

[6] Coursesteach, Understanding Matrices and Vectors: Essential Linear Algebra Concepts for Data Science-ML (Part 10), *Medium*, last accessed on 2024-12-25.

[7] Miles Lubin et al., JuMP 1.0: Recent improvements to a modeling language for mathematical optimisation, *Mathematical Programming Computation*, 2023.

[8] The Trucker News Staff, New study shows big rig operating costs reaching all-time high, *https://www.thetrucker.com*, 2023.