Volume : 4 | Issue : 10 | October 2015 • ISSN No 2277 - 8179

Route Optimization Using Opensolver

Engineering

KEYWORDS : Genetic Algorithms, Linear Programming, Microsoft Excel Solver, Network Optimization, OpenSolver, Route Optimization

Shubham Khanve Sardar Vallabhbhai National Institute of Technology, Surat

ABSTRACT

A big and complex model is formed and solved using 'OpenSolver' as an Excel Add-In. The model is regarding route optimization, which involves nine manufacturing cities and fourteen to be delivered cities. The real scenario of 'UltraTech Cement Ltd.' is taken. Optimization is performed under the constraints of limiting the producing capacity of manufacturing plants and limiting demand by demanding cities. This modelling is incredibly simple to deploy and cost-efficient and doesn't need any special skills.

INTRODUCTION

Finding an alternative with the most cost effective or highest achievable performance under the given constraints, by maximizing desired factors and minimizing undesired ones is called optimization of a given system. It can involve a different production methodology in an industrial plant, aiming to improve production within less time or to reduce the cost of a product to replace a certain type of constitutive material [1]. It also involves reducing the cost of operation by any means once the product was manufactured. One such type is transportation cost.

Some industries have few manufacturing plants and usually located far from each other. This is generally based on supply and demand created in a particular industry in a particular city. Demand of goods may be from different cities located homogeneously around cities having manufacturing plants. This optimization involves reducing transportation cost by optimizing distance to be travelled by transporters between 'manufacturing cities' and 'demand cities'. This problem is known as Route Optimization.

Route Optimization tells about 'demand cities' that from which 'manufacturing cities' they will receive goods. It may happen that it is surrounded by two or three 'manufacturing cities' equidistantly. The Question is from which 'manufacturing city' they should receive so that transportation cost is reduced. It's turned out to be a complex network as a number of 'manufacturing cities' or 'demand cities' increases. It is further complicated by having certain constraints. One of this is a different capacity of producing goods of different manufacturing plants. And even more complicated by the fact that each city has its own demand. The problem turned out to be a challenge both for practical applications and theoretical investigations [2].

Route Optimization problem belongs to a class of frustrated problem [3]. This paper is an attempt to simplify this problem using Microsoft Excel, 2013 (a proprietary spreadsheet software of Microsoft comes within Office 2013).The proposed method is very easy to deploy and can solve complicated optimization network problems based on similar lines.

OPENSOLVER

This paper uses an optimizer called 'OpenSolver'. This is an open source Excel add-in that allows spreadsheets users to solve their Linear, Integer, Non-Linear Programming models using COIN-OR CBC Solver [4]. This can be downloaded from 'opensolver.org' (free of charge) and is installed in Excel as per instructions given on the website. Excel also included a pre-installed Solver developed by Frontline Systems. But OpenSolver is capable of solving more complicated and larger models. OpenSolver model builder contains – Objective Cell, Variable Cells, and Constraints.

MODEL-DATA

Before beginning model, appropriate data is to be collected. This paper uses a real scenario of an industry. UltraTech Cement Ltd. is the largest manufacturer of grey cement, Ready Mix Concrete (RMC) and white cement in India [5].

Of the total, its nine manufacturing plants from seven states are considered. These are, Karnataka – Aditya Nagar (5 MT), Uttar Pradesh - Aligarh (1.3 MT), Punjab - Bhatinda (1.75 MT), Rajasthan – Dadri (1.3 MT) & Kotputli (3.1 MT), Gujarat – Jafrabad (0.5 MT) & Rajula (5.8 MT), Tamil Nadu – Reddipalaym (1.4 MT), and Maharashtra – Solapur (1.8 MT) [6]. Brackets show respective manufacturing capacity in Metric Tons (MT) of cement. It is to be delivered in capitals of fourteen states. These are, Sikkim – Gangtok (1.512 MT), Assam – Dispur (1.86 MT), Bihar – Patna (1.235 MT), Chhattisgarh – Raipur (1.735 MT), Goa – Panaji (1.433 MT), Gujarat – Gandhinagar (1.69 MT), Haryana – Chandigarh (1.435 MT), Karnataka – Bengaluru (1 MT), Madhya Pradesh – Bhopal (1.58 MT), Maharashtra – Mumbai (1.756 MT), Rajasthan – Jaipur (1.466 MT), Telangana – Hyderabad (1.796 MT), Uttar Pradesh – Lucknow (1.652 MT) , and West Bengal – Kolkata (1.8 MT). Brackets give city's assumed demand of cement. They are assumed such that total demand is equal to total production.

The distance of each 'manufacturing city' from 'demand city' is calculated using 'distancesfrom.com'. Figure 1 shows these distances in kilometres.

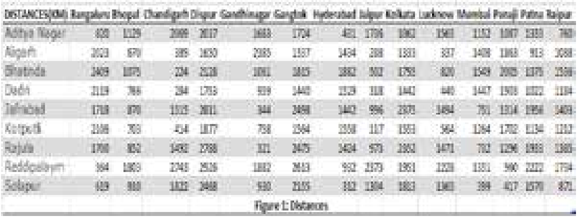
MODELLING

A. UNITS TABLE

The second Table is made for optimizing units (in tons) of cement to be taken from 'manufacturing city' to 'demand city'. As shown in figure 2, the last column is of 'Supply Available' which is manufacturing capacity of respective plants. The Last Row is of 'Demand', which is the total demand by a respective city.

Column – 'Supply Given' (Green Coloured): It is sum of supply given by 'manufacturing city' to each city. It is zero initially. This can be done by adding individual cells of a respective row or by using function 'sum'. For Example – '=sum(B13:O13)' or '=B13+C13+D13+E13+F13+G13+H13+I13+J13+K13+L13+M13+N13+O13'. This is repeated for every row.

Row – 'Given' (Pink Coloured): It is the sum of supply given to 'demand cities'. For Example – '=sum(B13:B21)' or '=B13+B14+B15+B16+B17+B18+B19+B20+B21'. This is repeated for every column.



DISTANCES(KM)	Bangalore	Bhopal	Chandigarh	Dispur	Gandhinagar	Gangtok	Hyderabad	Jaipur	Kolkata	Lucknow	Mumbai	Panaji	Patna	Raipur
Aditya Nagar	828	1129	2099	2817	1683	1724	421	1736	1062	1543	1152	1087	1333	783
Aligarh	2023	670	385	1650	2085	1337	1434	288	1331	337	1408	1863	913	1088
Bhatinda	1409	1075	224	2528	1061	1815	1882	502	1792	820	1549	2005	1375	1596
Dadri	2119	786	284	1783	919	1440	1529	318	1442	440	1447	1901	1022	1104
Jafrabad	1718	870	1515	2811	344	2488	1442	996	2371	1484	701	1314	1956	1409
Kotputli	2109	703	414	1877	758	2344	2358	117	1583	564	1264	1702	1134	1212
Rajula	1700	852	1492	2788	321	2475	1424	973	2352	1471	702	1296	1983	1380
Reddipalaym	364	1803	2743	2526	1882	2613	932	2173	1961	1728	1331	590	2222	1794
Solapur	419	910	1825	2468	930	2155	312	1304	1883	1340	399	417	1570	871

Figure 1: Distances

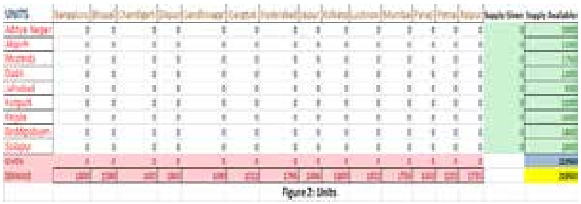

UNITS
Supply Given
Supply Available
Aditya Nagar
Aligarh
Given
Demand

Figure 2: Units

At ending of row and column: Yellow coloured cell signifies sum total of 'Demand' and Blue coloured cell signifies sum total of 'Supply Available'.

B. FINAL DISTANCES TABLE

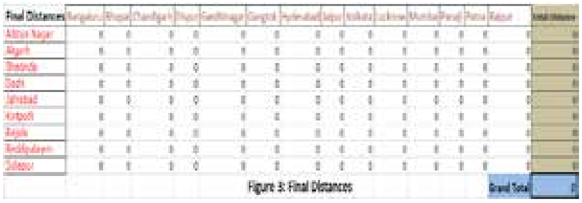

Final Distances
Total Distance
Aditya Nagar
Aligarh
Grand Total

Figure 3: Final Distances

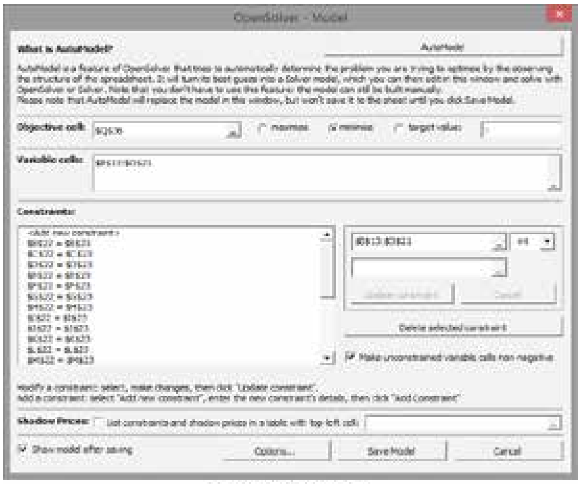
A third table is made for measuring 'Final Distances' i.e. actual distance to be travelled from respective 'manufacturing cities' to 'demand cities' as shown in figure 3. Each cell uses a function 'IF'. For Example, a cell corresponding to Aditya Nagar and Bangaluru uses '=IF(B13=0,0,B2)'. This is read as 'IF the value stored in B13 cell is zero then make this cell zero otherwise make this cell equal to the value stored in B2'. The value stored in B13 is Units (Figure 2) value corresponding to the same pair of the city. And the value stored in B2 is Distances (Figure 1) value corresponding to the same pair of cities. Last column (Light Brown Coloured) signifies the total distances by respective 'manufacturing city'. This uses function 'sum' or individual corresponding cells are added. 'Grand Total' (Blue Coloured) gives us the sum of all cells in 'Total Distance'. Each Cell is initially set to zero.

In this figure, every cell is assigned to 'IF' function because we have to consider distance only when a particular 'manufacturing city' is supplying to a particular 'demand city'. If that is not done, we are taking value zero, but if that is done, then we are taking its distance from figure 1 so that it will come into account.

C. BUILDING OPENSOLVER MODEL

We have to minimize 'Grand Total' so that least possible distance is to be travelled from 'manufacturing city' to 'demand city' such that each 'Demand City' fulfilled its demand and each 'Manufacturing City' is able to provide its available supply. Open Data in Excel and then click on Model under OpenSolver. A screen like figure 4 will appear.

Under Objective Cell, select 'Grand Total' Cell. Objective Cell is the cell for which we are modelling. This cell can be maximized, minimized or can be set to a target value. In this model, we want to minimize objective cell thus select 'minimize' in OpenSolver Model. Next we have to select our variable cells, i.e. those cells value which we want to change. In this case, our variable cells will be all cells under Units table (figure 2). Select all cells under units table (except the name of cities, only a cell whose value is a number).



AutoModel is a feature of OpenSolver that tries to automatically determine the problem you are trying to optimise by the observing the structure of the spreadsheet. It will turn its best guess into a Solver model, which you can then edit in this window and solve with OpenSolver or Solver. Note that you don't have to use this feature: the model can still be built manually.
Please note that AutoModel will replace the model in this window, but won't save it to the sheet until you click Save Model.

Modify a constraint: select, make changes, then click "Update constraint".
Add a constraint: select "Add new constraint", enter the new constraint's details, then click "Add Constraint".

Figure 4: OpenSolver Model

Now, we have to add constraints. A constraint is a limitation or restriction in a model, under which a model have to optimize. In our model, we have three generalized constraints. First one is that in Figure 2, our 'Supply Given' should match with 'Supply Available'. Second is that 'Given' should match with 'Demand' in figure 2. And third is that since we have taken integers, each value should also be integer in figure 2. So in OpenSolver Model, we have to add these constraints. Under Constraints, click first (top & left-hand side) blank. Click on any cell from 'Demand' from figure 2. Then select next blank as '=' and then below blank as cell from corresponding 'Given' (from Figure 2). Similarly, do it for all 'Demand' and 'Given' combinations. This completes our one constraint. Do this similarly for 'Supply Given' and 'Supply Available' combinations (from Figure 2). This completes our second constraint. For the third constraint, select all cells (excluding city names itself) in the first blank and then 'int' in the second blank and keep third blank as it is. Click on save model.

RESULTS AND DISCUSSIONS

You can verify your model by clicking on 'Show/Hide Model' under OpenSolver in DATA Tab. To solve model, click on 'Solve'. You can check that values in Figure 3 and Figure 4 tables have changed. OpenSolver uses its inbuilt algorithms to solve this particular problem. It has minimized the value of 'Grand Total' to its fullest possible.

Figure 2 Table will provide you about number of units to be taken from particular 'manufacturing city' to shown 'demand city'. There will be lot of zeroes that means that particular 'manufacturing city' should not have to bother about zero shown 'demand city'.

Figure 3 Table will provide you about the corresponding distances to be travelled. It is already minimized thus company should not bother about any decision related to distances.

Minimization in transportation is thus done, which will ultimately reduce transportation cost. This is simplest and very easy method to deploy. Company usually pays a big amount of money to route optimization providers, but, however, this model use OpenSolver which is free of cost. This model also does not require any special skills and can be performed by anyone. This route optimization can also be performed by built-in add-in 'Solver'. However, it asks for selecting a solving method. Simplex LP will unable to perform it. GRG Nonlinear might do it, depends on your size of the model. Evolutionary method is the best-applied method. It uses genetic algorithm to solve a particular model. However, OpenSolver is more easy and powerful to use.

REFERENCE

[1] Walliston S. Fernandes, Valerio S. Almeide, Francisco A. Neves, Marcelo Greco. 2015. Topology optimization applied to 2D elasticity problems considering the geometrical nonlinearity. [2] Ebeling, W. 1990. Applications of Evolutionary Strategies, Syst. Anal. Model. Simul. [3] Frank Schweitzer, Helge Rose, Werner Ebeling. 1997. Optimization of Road Networks using Evolutionary Strategies. [4] Andrew J Mason. 2012. OpenSolver – An Open Source Add-in to Solve Linear and Integer Programmes in Excel. [5] Overview. Retrieved from http://www.ultratechcement.com/about-ultratech-cement-ltd. [6] Company wise Installed Capacity, Production Capacity of Cement Plants. 2012. Department of Industrial Policy and Promotion. Ministry of Commerce & Industry, Government of India.