energysys

Complex problems made simple

Route optimisation using Dijkstra's algorithm

EnergySys
Hudson House, 8 Albany Street,
Edinburgh
EH1 3QB
t: +44 1224 433 493
e: sales@energysys.com
www.energysys.com

Table of Contents

1   Abstract

Every day, businesses face complex operational decisions. Often, these need to be made by those on the ground, running the business day-to-day. They must analyse and solve operational problems as they occur, make quick well-informed decisions, and, importantly, gain or maintain competitive advantage.

If you are one of those people (engineer, analyst or technologist) then you need tools that help support these activities and are quick and easy to use. Your systems need to be as flexible and agile as you.

EnergySys is a low-code cloud-native platform that is agile, effective, and rapidly deployable. It enables business people like you to configure applications to suit their particular requirements, resulting in a solution that yields to the business as opposed to the traditional approach of the business bending to fit the technology.

In this paper we aim to demonstrate to you just how easy this is. We take a simple example and walk through the process of configuring EnergySys to support the business in finding the most 'efficient' route across a network, in this case a pipeline network, using Dijkstra's algorithm1. This is an iterative process which is quite difficult to implement fully in Excel but is made much simpler by the way the calculation mechanism in EnergySys utilises blocks of logic.

It reflects a practical problem facing pipeline operators every day, but it is also a great example of how an algorithm can be implemented and deployed to an application by the business, for the business. Contrast this with typical gas management solutions, which might use this type of algorithm. They can be unwieldy and costly, requiring specialist skills to implement, customise and maintain. The result is a brittle and inflexible application with high ongoing total cost of ownership (TCO) to the business.

2   Background

2.1   Transmission Factors

Pipeline operators are required to manage many operational and commercial factors when seeking to run their network optimally. It is a complex problem and involves continuously juggling competing requirements, with limited resources, in an environment of continual change. Calculating the most "efficient" route for product through the pipeline network is just one example of the kind of decisions that need to be made every day.
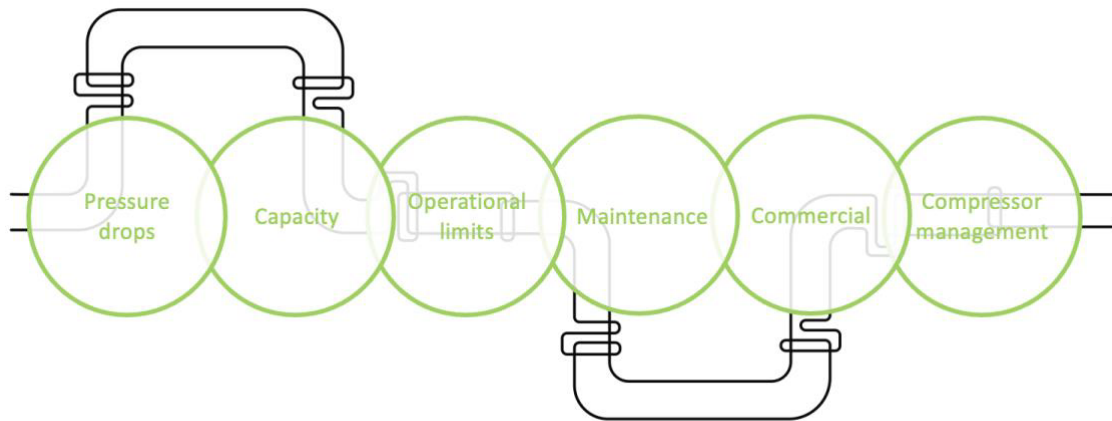
Figure 1 Example Pipeline Operation Factors

The following are typical factors that need to be considered and managed.

1. **Pressure Drop**, attributable to the frictional resistance of the pipe wall and other pipeline components.
2. **Capacity** (or line pack) within the network which, for compressible fluids, varies with pressure.
3. **Operational Limits** based on the nameplate characteristics of the pipeline.
4. **Maintenance** both planned and unplanned impacting the delivery of the transmission service.
5. **Commercial** factors such as rights to capacity and the financial impact of operational decisions.
6. **Compressor Management** to ensure that compressors are activated when they are needed, for the duration they are needed, producing the output required.

Managing these factors is not possible on whiteboards or excel and often enterprise solutions are costly, time consuming to implement and then require extensive ongoing management just to keep up with the business. But, what if a solution existed that allowed you to take advantage of the flexibility of a whiteboard and excel yet provided you with the integrity and power of an enterprise application?

Let's take a look at a specific scenario to demonstrate how EnergySys would address such an issue.

3   The Scenario

3.1   Nodes and Pipeline Sections

Imagine a bi-directional pipeline network, consisting of nodes (receipt, storage and/or delivery points) and connecting vertices (pipeline sections). Each pipeline section has a start and an end node, however not all nodes are currently connected to the network.

The pipeline connecting two nodes has a 'weight' associated with it. This has been defined by the business and represents an index number derived from continual analysis of the operational factors detailed above.
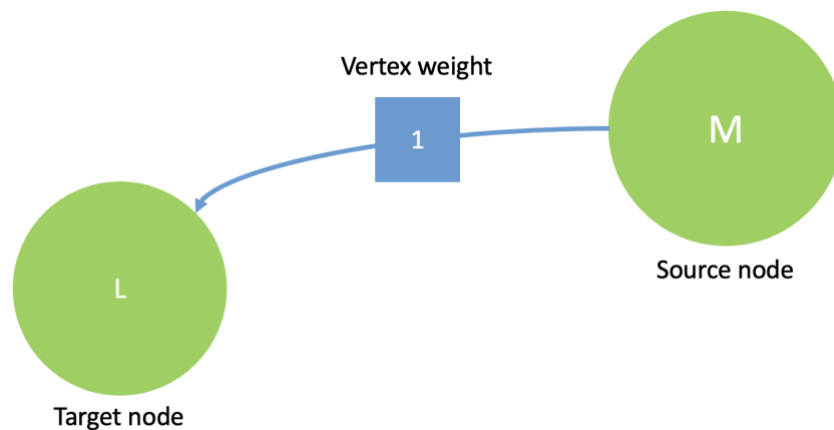
Figure 2 Sample Node and Vertex with Weighting

Within our network, our pipeline operative is tasked with moving gas from node A to node K, across the network and needs to know the optimal route to use from a selected entry point to a selected delivery point. The ability to store the gas at intermediate nodes is also a consideration and so the routing algorithm will consider all nodes as potential targets.

The weighting across each section is continually changing throughout the gas day and the algorithm needs to be performant in terms of execution time and accessibility of results.

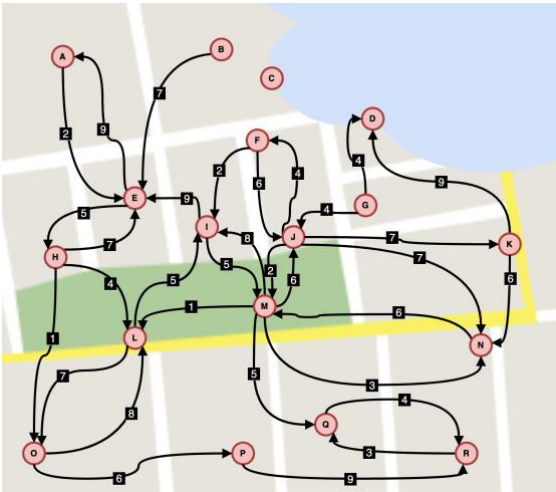The network can be displayed both geographically and logically as shown below.
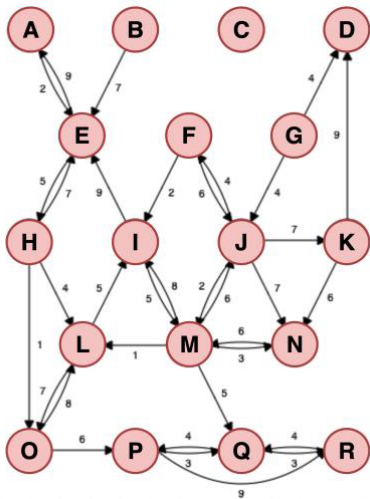
Figure 3 The Geographical Network

Figure 4 The Logical Network

4   **Configuring The Algorithm**

Djikstra's algorithm can be used to solve this kind of network problem and is a great example of the kind of analysis that can be achieved in EnergySys. At a high level, the algorithm identifies a starting point, analyses all connected nodes that have not been assessed previously, identifies the node with the lowest 'weight' and then uses that as the starting point for further analysis.

The process then repeats until all nodes have been analysed.

EnergySys provides the end user with the capability to configure objects (data tables) to store data and processes for performing calculations using this data. The process logic is defined using Excel standard functions and formulas. Configuration is quick and easy and hugely flexible.

4.1 EnergySys Object Configuration

Four objects were identified as being required and are created in EnergySys. These objects are;

1. Node – Contains the list of nodes, a description for each one and their longitude and latitude.
2. Visited – Contains a list of the nodes that have been visited during the algorithm's execution.
3. Network – Contains the relationship between nodes and the weighting assigned to each connecting pipeline section.
4. Result – Contains the results for each execution of the algorithm as the network is traversed.
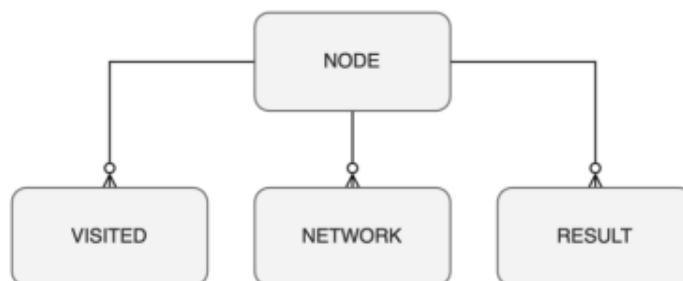
Figure 5 Object Structure

Now that the objects have been created, and their associated screens generated attention can turn to configuring the algorithm.

4.2 EnergySys Process Configuration

4.2.1 Process Overview

The objects are used, along with the algorithm logic held in the process workbook, to construct a process chain, or workflow. Two processes are identified as being required. These processes are;.

1. Process 10 – Receives the user's chosen Start Node parameter, organises, and then initiates the workflow.
2. Process 20 – Applies Djikstra's algorithm to a single node, using logic defined by simple Microsoft Excel functions within the workbook to identify the most efficient route across the network. Calls the next node in the iterative sequence until all nodes have been processed.

The use of Microsoft Excel and EnergySys' powerful calculation engine allows the user to benefit from spreadsheet simplicity, handling a single node at a time.
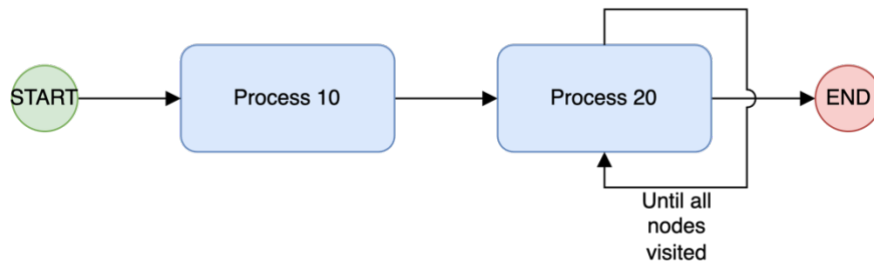
Figure 6 Process Chain Overview

5   Process Execution

5.1   Executing Processes

There are several options when deciding how to execute a process within the EnergySys Cloud platform. In this example a manual trigger has been selected, in reality an event driven, or schedule driven, trigger would likely be used to start the algorithm.

5.2   Process 10

Manually triggered, Process 10 brings in the NODE and NETWORK objects as inputs to the process, along with the user's chosen Start Node parameter. These are then consumed by the calculation engine using logic defined in MS Excel and records stored back into the database to the VISITED, RESULT and RUN PROCESS objects, the latter forming the chain to Process 20.
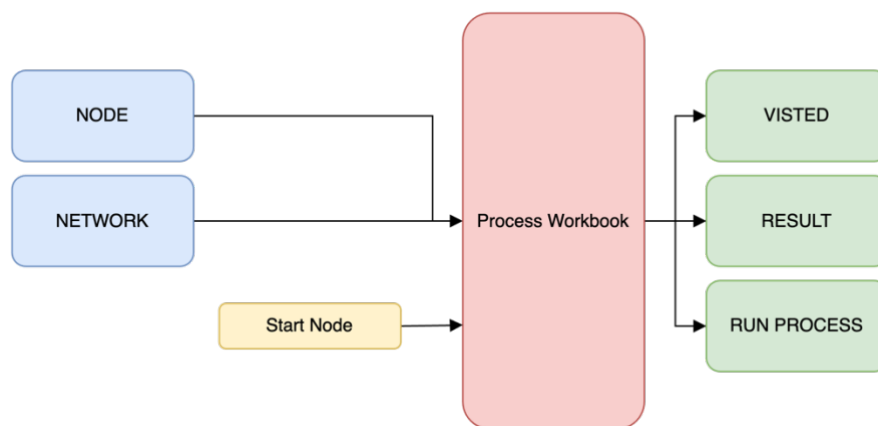
Figure 7 Process 10

5.3   Process 20

Automatically triggered by Process 10, Process 20 starts the task of working through the network as defined in the NETWORK object. Using the process workbook logic, it is able to define the Node it is currently analysing (Current Node), the current round of analysis (Current Round) and the previous round of analysis (Previous Round).
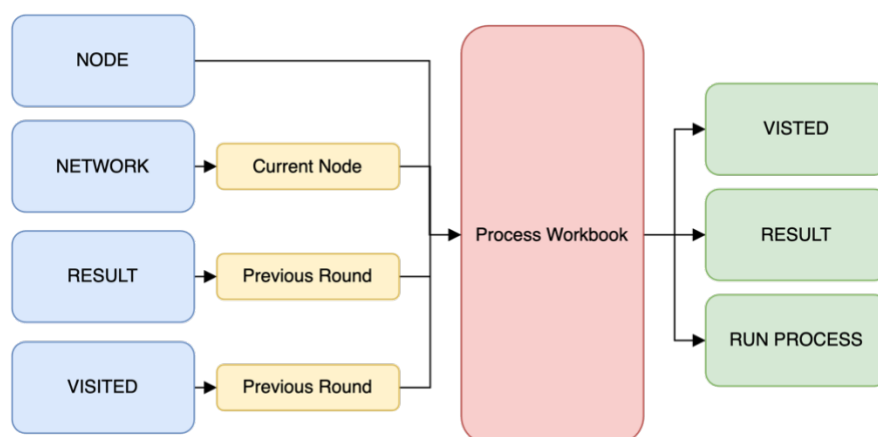
Figure 8 Process 20

The input objects are filtered to limit the data being presented to the algorithm, this increases process performance and makes the subsequent logging clearer and more user friendly.
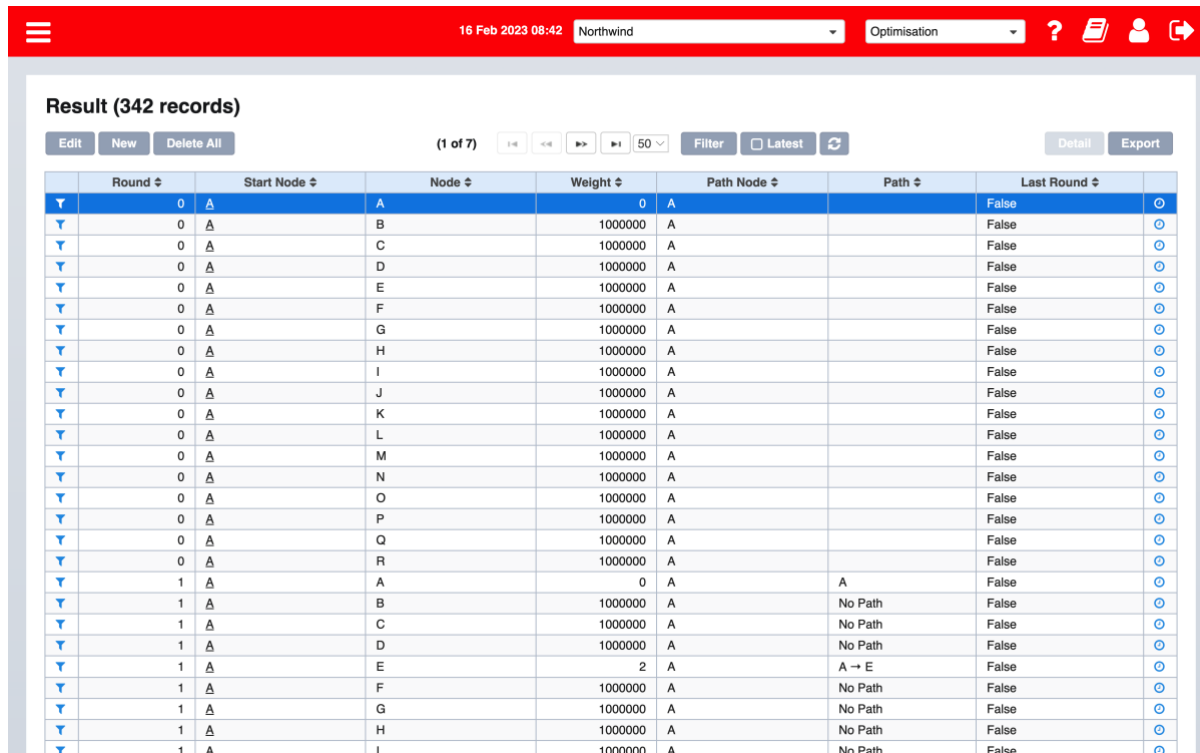
All iterations (or rounds) of the network are recorded, so that at any stage of the algorithm the user can review the decisions made by the EnergySys platform.

6   Results

6.1   The RESULT Screen
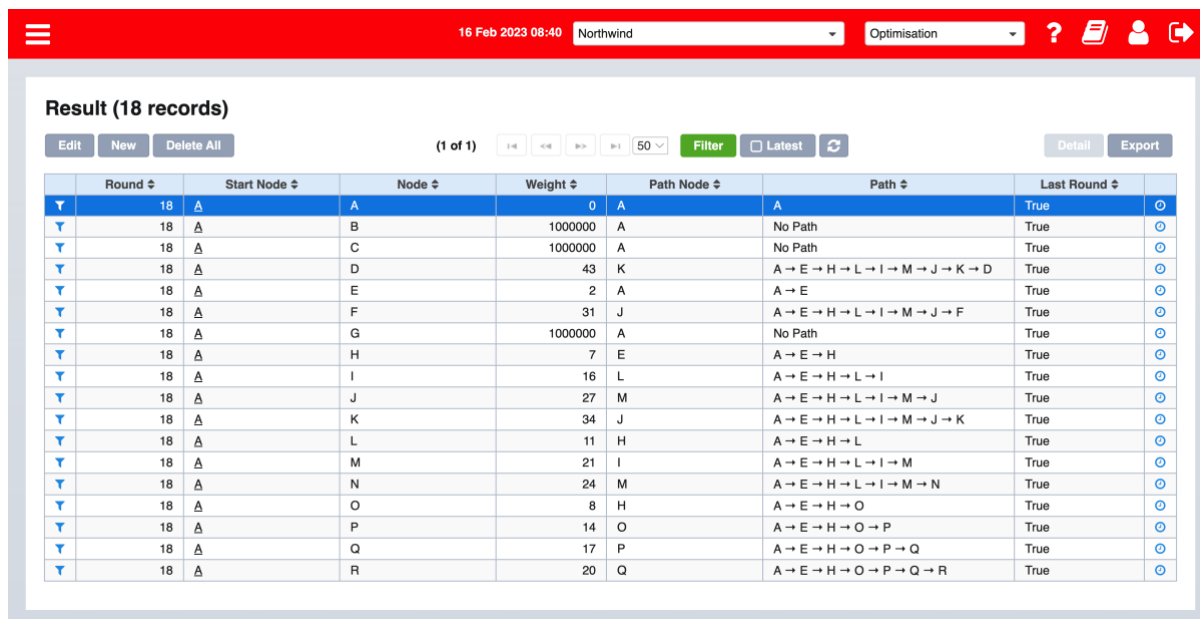
The results of the process can be seen through the application screens. The example screen below displays the algorithm results at the end of each iteration. As the workflow executes it is possible to see the shortest pathway being identified and built up in the Path column.

Figure 9 The RESULT Screen

When the workflow is complete, the screen displays the final results of the algorithm:

Figure 10 the Filtered RESULT Screen

6.2　The Calculation Log

All executed process within EnergySys have an associated calculation log. This log contains all the details behind the algorithm. It clearly displays the inputs to the calculation, the logic applied and the outputs stored into the database.
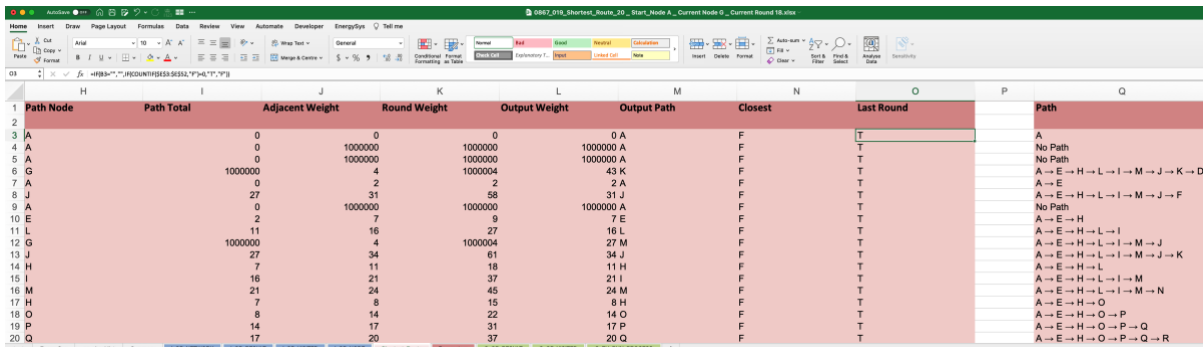
The calculation logs not only provide a complete and easily readable audit trail of the logic applied, but also allow users to undertake offline scenario modelling.

Figure 11 Sample Calculation Log Files

Opening any of the files in the log allows the user to clearly follow the logic, completely removing the traditional issue of black box solutions holding logic that can only be created and monitored by costly consultants who have little knowledge or regard for the industry in which you operate.

Figure 12 Inside a Sample Calculation Log File

6.3　Other Reporting Options

Although not included in this paper, it is possible to easily generate the results of the algorithm in .xlsx, .txt, .csv or .pdf formats.

In addition, EnergySys' O-Data compliant web services API allows business intelligence tools like Power BI and Spotfire to access live data from within the database. Recording the longitude and latitude against each node allows geographical maps of the network to be created to assist with analysis and visualisation.

7   The Most Efficient Route

Studying the algorithm, the results tell us that the most efficient path from node A to node K, for our example, comes with a weighting of 34 and is as follows;

Round ⇕	Start Node ⇕	Node ⇕	Weight ⇕	Path Node ⇕	Path ⇕	Last Round ⇕	
18	A	A	0	A	A	True	
18	A	B	1000000	A	No Path	True	
18	A	C	1000000	A	No Path	True	
18	A	D	43	K	A → E → H → L → I → M → J → K → D	True	
18	A	E	2	A	A → E	True	
18	A					True	
18	A					True	
18	A					True	
18	A	I	16	L	A → E → H → L → I	True	
18	A	J	27	M	A → E → H → L → I → M → J	True	
18	A	K	34	J	A → E → H → L → I → M → J → K	True	
18	A	L	11	H	A → E → H → L	True	
18	A	M	21	I	A → E → H → L → I → M	True	
18	A	N	24	M	A → E → H → L → I → M → N	True	
18	A	O	8	H	A → E → H → O	True	
18	A	P	14	O	A → E → H → O → P	True	
18	A	Q	17	P	A → E → H → O → P → Q	True	
18	A	R	20	Q	A → E → H → O → P → Q → R	True	

A -> E -> H -> L -> I -> M -> J -> K

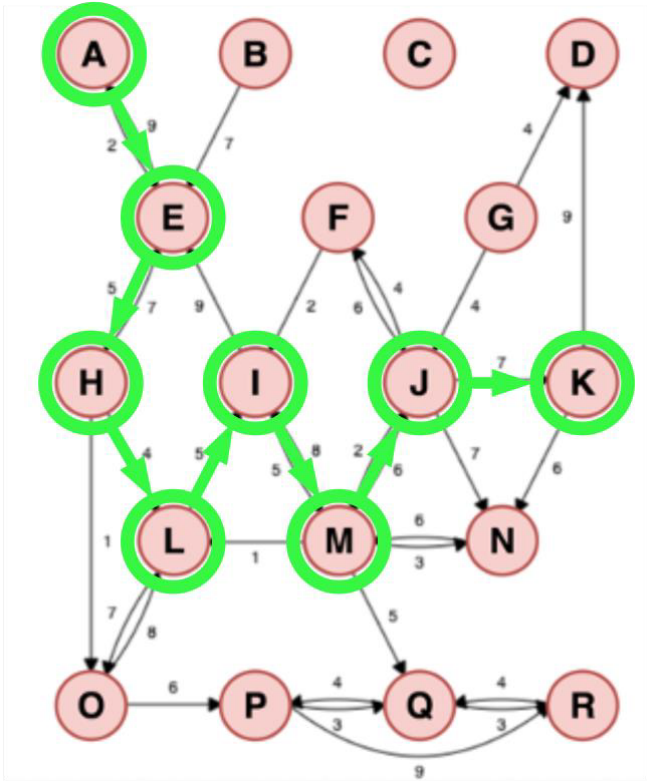*Figure 13 The Most Efficient Route (Screen View)*

Figure 14 The Most Efficient Route (Logical Diagram)

8   Conclusions

This scope of this paper is deliberately limited; the intent is to highlight just how easy it is to configure a technical application in EnergySys and how powerful the calculation engine can be.

Of course, there are many extensions and enhancements possible to this algorithm if required by the business. For example, if maintenance events, nomination forecasts and commercial data were stored in the EnergySys platform then these could be used within the logic. Notifications could be configured allowing key business users to be notified of potential issues on the network in advance of them occurring.

This algorithm is also typical of the type of calculations undertaken for emissions management. For example, it could be configured to enable routes to be selected based on their associated environmental impact, thus enabling an organisation to assess operational and commercial decisions with environmental impacts.

EnergySys also offers integration with partner products, algorithms and template applications providing a huge resource for engineers, analysts and technologists wanting to build their own technical enterprise-scale applications.

9   EnergySys

The EnergySys Cloud Platform is fast being recognised as the de-facto solution for data centric technical organisations.

It has been designed, from the ground up, to make it possible for business people to create secure, resilient and scalable technical applications that exactly meet their needs.

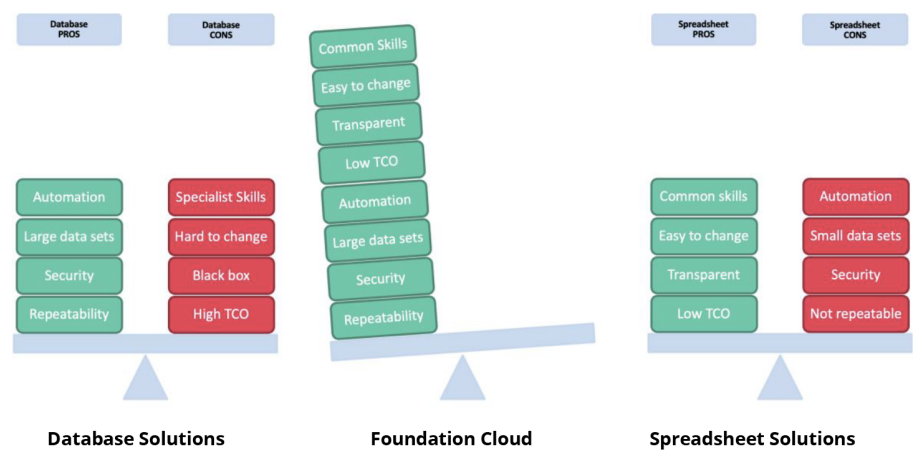It provides spreadsheet simplicity with database integrity, allowing users to benefit from a solution that they can configure, manage and understand.

Figure 15 EnergySys - The Best of Both Worlds