APPLICATION OF GRAPH THEORY TO RESOURCE DISTRIBUTION POLICY-BASED SYNTHESIS OF INDUSTRIAL SYBMIOSIS NETWORKS

Sidanth Dayal

[BSc. Chemical Engineering, UKZN]

A dissertation submitted in fulfilment of the requirements of the degree

Masters of Science (MSc.) in Chemical Engineering,

College of Agriculture, Engineering and Science

University of KwaZulu-Natal

May 2021

Supervisor: Professor Randhir Rawatlal

As the candidate's supervisor I agree to the submission of this dissertation:

____

Prof. R. Rawatlal

DECLARATION 1 - PLAGIARISM

I, **Sidanth Dayal,** declare that

1. The research reported in this dissertation, except where otherwise indicated, is my original research.
2. This dissertation has not been submitted for any degree or examination at any other university.
3. This dissertation does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

a. Their words have been re-written but the general information attributed to them has been referenced

b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.

5. This dissertation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed

____

Date: 16/05/2021

DECLARATION 2 - PUBLICATIONS

DETAILS OF CONTRIBUTION TO PUBLICATIONS that form part and/or include research presented in this thesis (include publications in preparation, submitted, *in press* and published and give details of the contributions of each author to the experimental work and writing of each publication)

Publication 1:

Title: **Automated Industrial Symbiosis networks: Distribution Policy implications of the Hungarian method, Edmonds-Karp algorithm and the 2-Phase Simplex method.**

Authors:

1. Sidanth Dayal
2. Prof. Randhir Rawatlal

Sidanth Dayal conducted research and performed the experimental work (computer simulations) before writing the journal paper with Prof. Randhir Rawatal. The journal article is a condensed version of Chapter 4 through 8 of this dissertation. Prof. R. Rawatlal was the co-author of the journal article and is also the supervisor of the MSc degree.

Signed:

Sidanth Dayal

Prof. R Rawatlal

ACKNOWLEDGEMENTS

I would like to thank:

- My supervisor, Professor Randhir Rawatlal, for his assistance, guidance and wisdom during this research;
- The University of Kwa-Zulu Natal for providing me with a scholarship for this research and online resources;
- SASOL for providing me with financial support for this research;
- My colleagues Mr. U. Maharaj, Mr. P Biyela., Ms. K. Kalicharan, and the S.M.A.D.A research group for their support and insights; and,
- My parents and siblings for their motivation, consideration, and patience they provided during my research.

ABSTRACT

Industrial symbiosis (IS) involves the repurposing of waste and by-product streams from one chemical industry as feedstock to another. Given the growing environmental and economic concerns, it has become increasingly difficult for industries not to participate in IS. This has encouraged much research into the field, with IS network design being an important optimisation problem in the research space. However, challenges are associated with the creation of IS networks, with transportation costs and resource distribution being key factors. Furthermore, solution strategies are usually complex and neglect the structural features of the network.

A possible solution is the use of graph theory for IS network creation. It was hypothesized that structural features of an IS network can evaluate the effect of distribution policies on IS networks created by graph matching algorithms. The Simplex method (SM), Edmonds-Karp algorithm (FF), and the Hungarian method (HM) were adapted to model IS networks, with the intention to establish a ranking in the suitability in creating IS networks. The adaption rendered the algorithms applicable to feasible IS network discovery under different distribution policies.

This graph-based approach allowed for the seamless extraction of the network features as graph metrics. Rigorous testing of the adapted algorithms' performance using graph metrics was done by simulating numerous IS scenarios. It was found that HM identified connections that, on average, minimised transportation costs to the greatest extent. The HM created networks with the smallest travelling distance than those of SM and FF, showing a 9 % and 6.06 % lower value than SM and FF, respectively. Furthermore, HM-IS networks created more stable and fair networks, which was inferred from the graph metrics.

To confirm the HM's apparent superiority in IS network creation, a case study was simulated with the defined distribution policies being modelled from the matching features. Each distribution policy was quantified as a cost from which it was found that HM-IS networks had a 72.5 % and 74.9 % lower overall distribution cost than FF-IS networks and SM-IS networks, respectively. It was concluded that HM is the most suited for IS network creation and that graph-based modelling of IS is a feasible approach.

Contents

DECLARATION 1 - PLAGIARISM ii

DECLARATION 2 - PUBLICATIONS iii

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF FIGURES viii

LIST OF TABLES x

1 INTRODUCTION 1

2 LITERATURE REVIEW 10

2.1 Industrial symbiosis Implementation models 10

2.2 Common optimisation methods 12

2.2.1 Pinch Analysis 12

2.2.2 Mathematical programming (MP) 13

2.3 Other optimisation strategies 15

2.4 Decision-support tools and commercial software 16

2.5 Post-optimality decision support tools 17

2.5.1 LCA: Environmental objectives 18

2.5.2 Game theory: Economic objectives 18

2.5.3 Social network analysis: Social objectives 19

2.6 Matching algorithms 20

2.7 Summary 23

3 THESIS OBJECTIVES 25

4 MODELLING IS NETWORKS AS GRAPHS 28

4.1 Distribution policy 28

4.2 IS Problem Formulation 29

4.3 Graph modelling of the problem formulation 30

5 MATCHING INDUSTRIES BASED ON DISTRIBUTION POLICIES 33

5.1 The transportation problem and the 2-Phase Simplex method (SM) ... 33
5.1.1 A gentle introduction to linear programming ... 33
5.1.2 The transportation problem ... 36
5.1.3 Adaption of STP and SM to Industrial symbiosis ... 40
5.2 Maximal Flow Problem and The Edmonds-Karp algorithm (FF) ... 41
5.2.1 An introduction to the Maximal flow problem and the Edmonds-Karp algorithm ... 41
5.2.2 Adaption of the Edmonds-Karp algorithm to IS ... 45
5.3 Optimal Assignment Problem and the Hungarian method (HM) ... 48
5.3.1 Introduction to the Optimal Assignment problem and the HM ... 48
5.3.2 Adaption of the HM to IS ... 51
5.4 Dijkstra's algorithm and the Minimum Spanning tree ... 53
6 IS NETWORK GENERATION USING THE ALGORITHMS ... 55
7 METHODOLOGY BASED ON GRAPH METRIC ANALYSIS ... 61
8 SIMULATION RESULTS ... 66
8.1 General trends observation ... 68
8.2 Statistical graph metric results ... 72
8.3 Discussion ... 75
8.3.1 Main findings ... 75
8.3.2 Determination of Empirical Time complexity ... 79
8.3.3 Implications to Distribution policy ... 82
9 CASE STUDY: APPLICATION OF GRAPH-BASED ALGORITHMS FOR INDUSTRIAL SYMBIOSIS NETWORK CREATION ... 88
9.1 Case Study: Plastic waste distribution ... 89
9.2 Modelling Distribution policies ... 91
9.3 Graph representations for each algorithm ... 97
9.4 Results ... 98
9.4.1 Matching results ... 98
9.4.2 Matching feature results ... 100
10 CONCLUSIONS ... 110

11 RECOMMENDATIONS 113
12 REFERENCES 114
APPENDIX 122
APPENDIX A: Flowcharts for the Algorithms 122
APPENDIX B: NOMENCLATURE 130

LIST OF FIGURES

Figure 1-1: The Industrial Symbiosis of Kalundborg, Modified from Chertow and Portlock (2002). ... 2
Figure 1-2: A basic example of an undirected graph (left) and a directed graph (right). 5
Figure 1-3: Konigsberg bridge problem as shown as a town (a), schematic representation (b) and graph theoretic representation by Euler (c), as extracted from the paper by Amaral and Ottino (2004). 5
Figure 1-4: The social network structure and betweenness centrality of different firms in the Guijao IP, extracted from the paper by Song et al. (2018). 6
Figure 1-5: An example of a bipartite graph $G = G\ (U \cup V, E)$. 8
Figure 1-6: Conversion of an IS problem (as a bipartite graph) to a flow network for the application of the maximal flow problem. 8
Figure 4-1: Generalized problem formulation. 29
Figure 4-2: A generalized illustration of a bipartite graph G, showing two matchings. 31
Figure 5-1: Geometric representation of simple linear program. 34
Figure 5-2:Graph-theoretic representation of the Transportation problem for an m×n system. 37
Figure 5-3: An example of a flow network, G. 43
Figure 5-4: The flow network G (left) consisting of 1 arc with flow value $f12$, and the corresponding residual graph Gf (right), with two arcs determined from arc $e12$ of G. 43
Figure 5-5: An example of a flow network for which the BFS is more suited than DFS. 45
Figure 5-6: Graph representation of IS in maximal flow problems. 47
Figure 5-7: Bipartite graph showing a cycle C (blue edges) and a matching μ (highlighted edges). ... 49

Figure 5-8: BFS tree (left) for the bipartite graph (right) with matchings (red graph edges) and tight edges (blue dashed lines). 51
Figure 5-9: Solution output from the Hungarian method corresponding to the optimal entries in matrix B 53
Figure 5-10:MST for a 3 by 3 IS system. 54
Figure 6-1: The minimum spanning tree for the 6× 6 system. 55
Figure 6-2: Industrial symbiosis network rendered by the Hungarian method. 57

Figure 6-3:Industrial symbiosis network rendered by the Edmonds-Karp method. 57
Figure 6-4: Industrial symbiosis network rendered by the Simplex method. 57
Figure 6-5: IS network rendered by the Edmonds-Karp algorithm with a different order of the rows and columns in the assignment matrix. 58
Figure 6-6: Illustration of the change in order in a product attributes list. 58
Figure 6-7: Illustration of subgraphs in a graph G. 60
Figure 7-1:Flow diagram of the generalised framework implemented in the study. 63
Figure 7-2:Flowchart of the overall methodology used in this study. 65
Figure 8-1:Scaled Graph metrics for the **GROUP C** (left) and **GROUP S** (right) datasets against the difference between the number of suppliers and consumers. 69
Figure 8-2:Scaled Graph metrics for the **GROUP SC** datasets against the total number of suppliers and consumers. 70
Figure 8-3:Z-score normalized graph metric results for the **GROUP SC** datasets against the number of suppliers and consumers. 71
Figure 8-4: Illustration of the post-processing steps of the graph metric results. 75
Figure 8-5: An example of a tie in the chosen augmenting path in the BFS routine for the Edmonds-Karp algorithm. 77
Figure 8-6: General structure of the Simplex table (ST) for Phase 1 of the Simplex method. 78
Figure 8-7: Graph showing the runs to determine the time complexity of each algorithm. 80
Figure 8-8: Regression on the SM time complexity datapoints, showing an exponential trend. 81
Figure 8-9: Simple IS system for EVC demonstration. 85
Figure 8-10: IS networks for the HM (left), SM (right) and FF (centre) with different EVC profiles. 86
Figure 9-1: Geographical location of the suppliers and consumers of plastic waste. 90
Figure 9-2: Example of a supply chain for a manufacturing company, extracted from Spekman et al. (1998). 91
Figure 9-3:IS network (left) and matching M as a bipartite graph (right) from the HM algorithm. 99
Figure 9-4: IS network (left) and the matching M as a bipartite graph (right) from FF algorithm. 99
Figure 9-5: IS network (left) and the matching M as a bipartite graph (right) from SM method. 100
Figure 9-6: The χ results for the HM (top left), FF (top right) and SM (centre), show the number of vertices whose EVC value fall within 1 standard deviation of the mean EVC value. 102
Figure 0-1: Flowchart of the Simplex algorithm. 122
Figure 0-2: Flow chart of the 2-Phase Simplex method. 123
Figure 0-3: Flowchart for the Edmonds-Karp implementation in maximal flow problems. 124
Figure 0-4: Flowchart of the Edmonds-Karp algorithm in IS for transportation cost minimization. . 125
Figure 0-5:Flowchart of the Hungarian method. 126
Figure 0-6: Flowchart for the Hungarian method implementation for IS. 127

Figure 0-7: Flow Chart for Prim's Algorithm. 128
Figure 0-8: Flow chart for Dijkstra's algorithm. 129

LIST OF TABLES

Table 6-1: Product attributes for the set of suppliers. 56
Table 6-2: Product attributes for the set of consumers. 56
Table 7-1: Table showing the 3 blueprints used for creating different sets of datasets, with each set having d datasets. 64
Table 7-2: Bounds placed on the values of the product attributes. 65
Table 8-1: Mean Z-score value for the average number of graph components. 72
Table 8-2: Mean Z-score value for the average number of graph edges. 72
Table 8-3: Mean Z-score value for the average total edge distance. 73
Table 8-4: Mean standard deviation in the graph metric results for each type of dataset and algorithm pair. 74
Table 8-5: Algorithm time complexity (seconds) for each type of dataset. 81
Table 8-6: Algorithm time complexity (seconds) for different runs of each type of dataset. 82
Table 8-7: Graph metric results for the IS networks rendered by each algorithm. 86
Table 9-1: Product attributes for suppliers of plastic waste. 90
Table 9-2: Product attributes for consumers of plastic waste. 90
Table 9-3: Weighting coefficients for each distribution policy. 96
Table 9-4: Values chosen for the unit conversion cost factors, which would otherwise be empirically determined. 96
Table 9-5: Differences in the terms for the cost function for each algorithm. 98
Table 9-6: Graph metrics for the IS networks from each algorithm. 99
Table 9-7: Matching feature results associated with the exchange of by-product material for each IS network produced by the algorithms. 102
Table 9-8: Results showing the matching feature costs from each algorithm. 103
Table 9-9: Cost of each distribution policy, in terms of its index, for each algorithm. 103
Table 9-10: Matching feature results for the HM. 104
Table 9-11: Matching feature results for the FF algorithm. 104
Table 9-12: Matching feature results for the SM. 105
Table 9-13: Mean graph metrics from each algorithm's IS network across the iterations. 105
Table 9-14: Mean distribution index across all iterations for each algorithm and distribution policy pair. 106
Table 9-15: Mean EVC vertex value for each iteration and algorithm pair. 107

Table 9-16: RMSE for the EVC values for each iteration and algorithm pair. 107

1 INTRODUCTION

There is a growing concern on the political and social landscape on the consequences economic development has had on the environment. This has resulted in sustainable development becoming a key feature in government policies (Gibbs, 2008). Such measures are necessary to be taken since, in the absence of regulation, individuals overexploit the environment to their own advantage (Stavropoulos et al., 2018) resulting in the depletion of natural resources. The concept of sustainable development has been the basis of a significant body of research into industrial development with initiatives being taken that are aimed at increasing business success and preserving the natural environment (Boix et al., 2015). While process optimisation has been implemented successfully in the past, with improvements made in terms of environmental performance and profitability, there exists a limit due to controllability and flexibility criteria (Marton et al., 2016). Hence, solutions are required that lead to greater environmental and economic sustainability. Efforts to reconcile economic, social and environmental concerns culminated into the emergence of Industrial ecology (Gibbs, 2008).

Industrial ecology (IE) may be defined as a multi-disciplinary research field that seeks to contribute to sustainable development by closing materials cycles and realising a paradigm shift in industrial activity (O'Rourke et al., 1996). Frosch and Galllopoulos (1989) were early proponents of IE and explained that a more integrated model should replace the traditional linear individual manufacturing processes. An industrial ecosystem is needed that adopts a system view to optimise the raw materials usage, energy usage and capital. IE is a broad field with several traditional methods established to the study and development of IE. A promising method is Industrial symbiosis, a subfield of IE that has been described as a viable approach for the improvement of companies' operational and environmental performance (Clift and Druckman, 2015) and is the focus of this study.

Industrial symbiosis (IS) allows traditionally separate industries in geographical proximity to exchange energy and material flow as a collective approach to competitive advantage (Chertow, 2000). The waste streams or by-products of one plant can be used as a raw material to another thereby increasing profitability and environmental sustainability. Industrial symbiosis finds its analogy in biological systems in which there exists a symbiosis between different species such that the relationship results in mutual benefit (Schwarz and Steininger, 1997) . It is a field that allows for a systematic approach to sustainable development. IS has been shown to be a strong proponent for the achievement of waste minimisation and resource efficiency without negatively impacting economic growth (Martin and Harris, 2018) . Since industrialisation plays a key role in long-term economic growth (Haraguchi et al., 2019), adopting IS presents itself as the middle path to ensuring that such a role remains viable without comprising the state of the environment (Neves et al., 2020).

IS has developed as a natural consequence to the economic and political pressure experienced by companies from all stakeholders. The idea of industrial relationships mimicking organic relationships found in nature was put forth by George Renner in 1947 (Zhang et al., 2015). The successful realisation of this idea came in the form of industrial exchanges in Kalundborg, Denmark, shown in Figure 1-1, where a group of companies decided to participate in resource exchanges due to the low availability of groundwater and has since allowed for the realisation of high levels of environmental and economic efficiency by the participants (Chertow and Ehrenfeld, 2012). Since then, many IS cases have developed over the years, especially in countries like China, Japan, UK, Algeria and USA, to name a few (Neves et al., 2020).

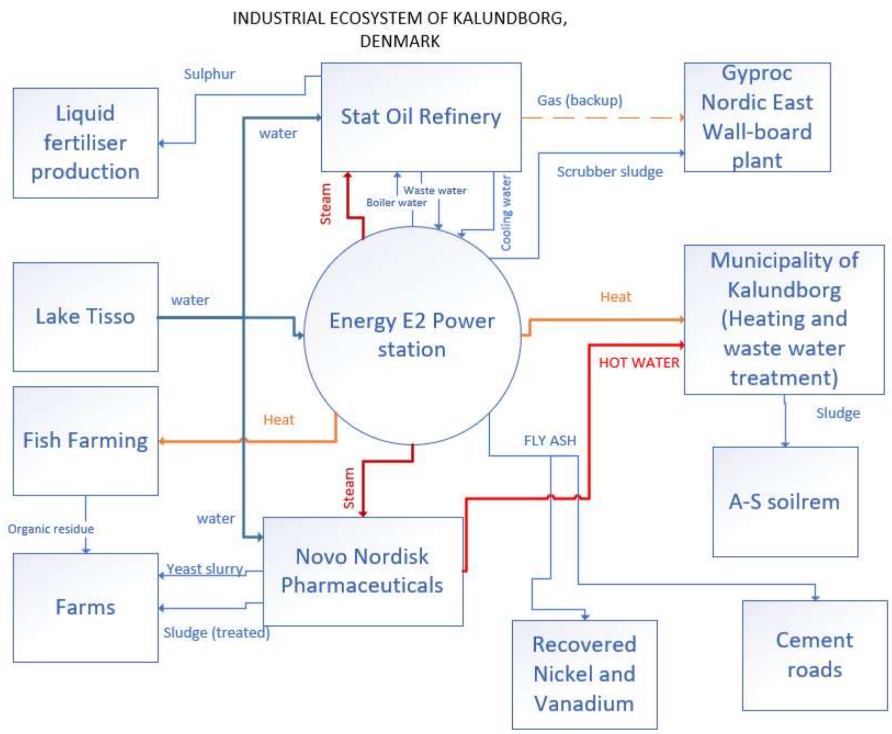


Figure 1-1: The Industrial Symbiosis of Kalundborg, Modified from Chertow and Portlock (2002).

This has encouraged much research into the field, with the design of Eco-Industrial Parks (EIPs) as an optimisation problem being a key research area. EIPs are industrial parks in which a collaboration occurs between the businesses in the park with the sole purpose of minimising material and energy waste through an exchange network of these waste streams. The result is the reduction in environmental impact without compromising profit (Kastner et al., 2015). Chertow (2007) has classified two models of symbiosis:

- Planned EIP model
- Self-organising symbiosis model

In a planned EIP model, there is an effort to identify companies from different industries before locating them together, whereas in a self-organised symbiosis model, companies are motivated to exchange waste streams and resources for business expansion and cost reduction, to name a few (Chertow, 2007). In both cases, close geographical proximity is a necessary precursor with some authors even stating it is critical to the success of IS (Christesen et al., 1999). Without incurring the cost of developing a planned EIP or limiting symbiosis to companies already in geographical proximity, Virtual Eco-Industrial Parks (VEIPs) can be created. VEIPs stretch beyond the traditional boundaries of IS whereby the exchange of waste material and energy occurs across industries that are not collocated (Zhang et al., 2015) .

When it comes to the identification of inter-company exchanges and design of IS networks, the research efforts can be split into two general approaches:

- Pinch analysis methods
- Mathematical programming optimisation

Pinch analysis consists of a set of methods that have been used in process industries to reduce energy demands and is based on thermodynamic principles. It is a set of systematic methods aimed at improving heat recovery energy use in process plants. Initially developed for heat integration in a processing plant, different variations were developed for different fields after analogies were created from the thermal pinch (Nemati-Amirkolaii et al., 2019). Mass and water pinch are the most successful extensions of the thermal pinch (Klemeš et al., 2018). Both mass and water pinch are concerned with the minimisation of scarce resources, such as water, while still ensuring that the processes operate in the most cost-efficient way.

The difference between the water and mass pinch is that while both are applicable to water minimisation problems, mass pinch is also concerned with the design of the mass exchange units (Isafiade and Short, 2019). Water pinch has been used to minimize freshwater usage and wastewater generation. The approach has been used extensively for the design of EIP networks. In these networks, inter-plant integration is achieved by the optimisation of water networks between the companies. The optimisation for these works involves the optimal distribution, treatment, and discharge of water between the processing units of the participating companies (Boix et al., 2015).

However, the vision of IS is far more extensive than this. Multiple objectives can be achieved in the creation of an IS, where the triple bottom line for all participating companies can be met. This, however, adds several layers of complexity with social, environmental, and economic objectives often conflicting (Boix et al., 2015). This type of complexity usually necessitates the use of mathematical programming optimisation techniques, which include Mixed-Integer Linear Programming (MILP), Non-linear programming (NLP), and Mixed-Integer-Non-linear programming (MINLP) (Kantor et al., 2020).

Mathematical programming, in this context, is a mathematical representation that reflects the aim of planning the optimal allocation of scarce resources. It is a well-developed branch of Operations Research (OR), a field that is concerned with the management of organisations by scientific methods (Hillier and Lieberman, 2010). The mathematical representation of the problem is a model which encompasses the constraints imposed on the problem as functional constraints and reflects the over-arching policy that governs the problem as an objective function equation. The model can solve maximization and minimisation problems by finding the global extreme value of the problem within the feasible region defined by the constraints of the problem.

Whether the functional constraints or objective function equation are linear or nonlinear determines whether it is a nonlinear program or not. Linear programs (LP) are those programs in which the variables are continuous, and both the constraint and objective function are linear. If some variables are restricted to integer values, then the model is said to be a mixed-integer linear program, requiring more complex solution strategies than its LP equivalent. The solution techniques to solving MILP centre around relaxations and the approaches listed below:

- Branch-and-bound methods
- Cutting plane methods
- A combination of the above methods

Relaxations involve the solving of the relaxed version of the problem, which is generally easier to solve, such that the solution provides a bound on the solution of the MILP. A common relaxation is the LP relaxation for MILP problems (Smith and Taskin, 2008). This involves solving the mathematical model, without imposing integer restrictions and using branch-and-bound methods and/or cutting plane methods to create subproblems until an optimal LP solution is rendered with all discrete variables taking on integer values. Most Premium MILP solvers, like CPLEX, use a combination of branch-and-bound and cutting plane techniques (Smith and Taskin, 2008).

All these methods that concern the design of an optimal IS network ultimately result in the creation of relationships between a set of suppliers and consumers. Depending on the nature of the symbiosis, the inter-relations can prove to be quite complex in terms of the number of connections and the exchange types (material or energy exchanges). The creation, analysis, and improvement of these complex inter-relations may be possible with graph theory due to its suitability in modelling systems that can be abstracted as a network (Ahmed et al., 2016, Mizuno and Ishida, 2016, Richardson and Thomson, 1996).

Graph theory is the study of graphs, which are mathematical structures used to represent pairwise relations between objects. The objects are represented as vertices (points) and edges are the lines/connections that define the relationship between the objects. There are many types of graphs, however, there are but a few relevant to this research.

Formally, a graph $G = G\ (V, E)$ is a set of vertices V connected by a set of edges E. The graph edges can either be undirected or directed (arcs), as shown in Figure 1-2. A simple graph is one that has no parallel edges or loops. Parallel edges connect the same vertices at their end points, whereas loops are edges with its endpoints attached to one vertex only. Vertices are said to be adjacent if they are connected by an edge/ directed arc.

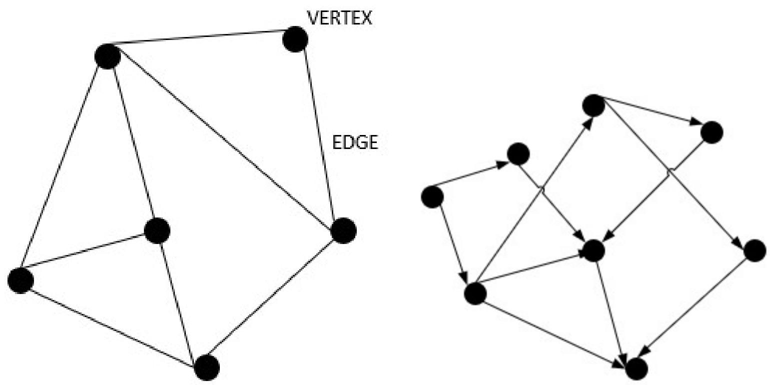


Figure 1-2: A basic example of an undirected graph (left) and a directed graph (right).

One can attribute the work of Leonhard Euler on the Königsberg bridge problem in 1736 as the initial literature of graph theory and the beginning of the field (Kruja et al., 2001). Ever since then, much study was devoted to investigating different types of graphs and their mathematical properties. For example, it was the work of Euler on the Königsberg bridge problem that resulted in his conclusion that there was no path that crossed each of the seven bridges exactly once, exposing the nature that the geometric position afforded the problem and resulted in the creation of Eulerian graphs as seen in Figure 1-3. A Eulerian graph contains a path, consisting of a sequence of graph edges, that starts and ends at a vertex u and contains every edge in the graph, with the set of traversed graph edges being distinct.

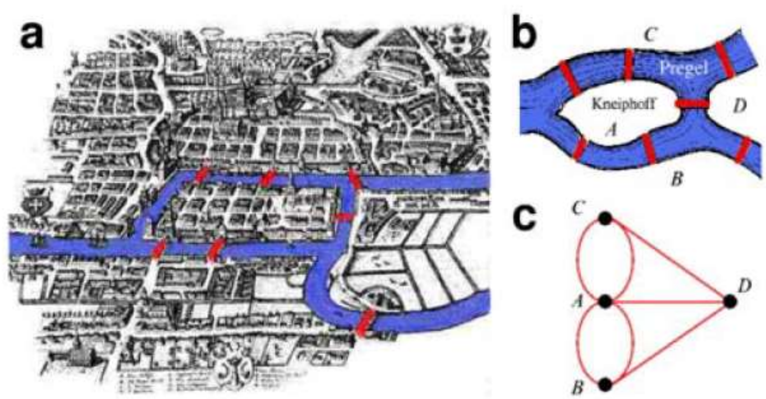


Figure 1-3: Konigsberg bridge problem as shown as a town (a), schematic representation (b) and graph theoretic representation by Euler (c), as extracted from the paper by Amaral and Ottino (2004).

A path can be defined as a sequence of vertices and edges, of the form $[v_{0i}, e_{ij}, v_{jk}, e_{kl}, ...]$ and describes the traversal of all edges in the graph with no edge being traversed more than once and no vertex being visited more than once, unless it is the starting vertex. Euler found that no such path exists in the case of the Konigsberg bridge problem.

For the case of industrial symbiosis, a graph G can be used to represent the IS, with companies being represented as the vertices V and the edges E used to represent the connections between them. After the design of the IS network with MP methods or pinch methods, a set of edges between the vertices of the graph represents the pairings made between companies. It would seem that the simplicity that graph theory affords in visualising IS networks is the reason that graph-theoretic modelling is commonly used in IS studies, either for facilitating the optimisation process using MP methods, or for providing descriptive and prescriptive analysis. The latter is made possible by Social Network Analysis (SNA), a popularised instance of network analysis, and has been used mostly to study the structural characteristics of IS networks (Fraccascia and Giannoccaro, 2020).

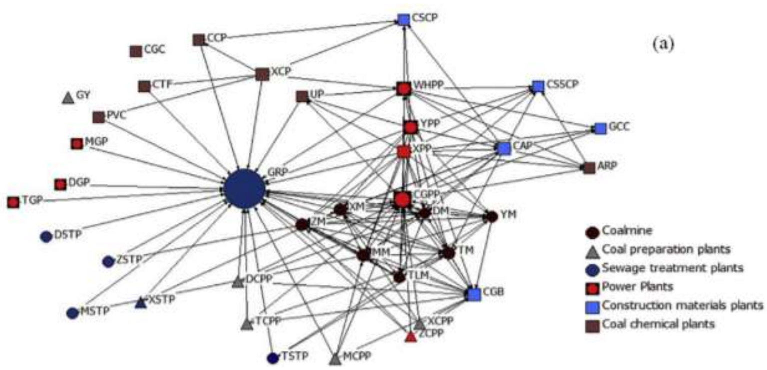


Figure 1-4: The social network structure and betweenness centrality of different firms in the Guijao IP, extracted from the paper by Song et al. (2018).

SNA studies are mostly focused on already implemented cases of IS and consists of quantitative measures used to formulate qualitative concepts related to the stability, resilience and robustness of networks. This is done using graph metrics, which are structural attributes extracted from the IS network, after it has been represented by a graph/network, as seen in Figure 1-4. The relationships that exist between the companies can be anything from material and energy flows to financial transactions (Schiller et al., 2014). Common metrics used can be summarised as either indicating “degree”, “closeness” or “betweenness” (Khakzad and Reniers, 2015), which are used in different extents to infer network stability, resilience, and ease of communication and has been used in engineering applications to assess cascading effects in process plants and hazardous environments (Khakzad et al., 2017, Khakzad and Reniers, 2015).

The growing interest and literature in graph theory naturally led to its introduction into its modern application of Operations Research (O.R). During World War 2 O.R was concerned with military management and focused on research into military organisations. O.R applies the scientific method for solving problems that are encountered by organisations. The problem in World War 2 was the difficulty in assigning available resources in a way that was most optimal to the organisation (the military) and this gave rise to O.R. This is similar to specific cases of IS, where the resource in question may be a scarce resource or energy and the task is to distribute the resource in the organisation (companies in the IS) in an optimal way.

Since the 1950's, its scope increased to the commercial and industrial sector (Hillier and Lieberman, 2010) with advancements made due to contributions by researchers and computing power. An important problem in OR is the network flow problem, which is concerned with the minimisation of the cost of moving materials through a network from the initial location of the materials to the locations where the demand for the materials exist (Vanderbei, 2020). Three specialised versions of the network problem are the transportation problem, maximal flow problem and the optimal assignment problem.

The transportation problem is concerned with determining the minimum transportation cost of distributing a single resource from a set of sources to a set of destinations. The transportation problem can be adapted to the case of IS, where the set of sources represent supplier companies of a resource, and the set of destinations represents that the consumer companies of the resource. The maximal flow problem is concerned with finding the maximum allowable flow of goods in a network from the starting location (source) to the destination (sink). This is relevant in the case of IS for situations where the maximal distribution of waste or resource material is required

The assignment problem can be explained in the form of an analogy. Given that there are m jobs and m employees, with each employee costing the company a certain cost of money for each job he/she is assigned to, find the optimal assignment of employees to jobs such that the total cost to the company is minimized. This is similar to IS, where the goal is to create optimal connections between suppliers (employees) and consumers (jobs) in such a way that the cooperative benefits received is maximised (cost of assignment is minimised). Not only are these three network flow problems relevant in the case of IS, but they also have strong graph representations. Of specific interest is the representation of these problems as bipartite graphs.

Bipartite graphs have been used to model the assignment problem and the transportation problem. A bipartite graph can be defined as a graph whose vertices can be split into two distinct sets and are connected by the set of graph edges such that adjacent vertices belong to a different vertex set. A bipartite graph is presented in Figure 1-5. The collection of edges that define the adjacent vertices is called the cut set $\bar{C}$. For the case of IS, this collection of graph edges may represent the connections and resource exchanges made between supplier companies, as represented by the set of vertices U in Figure 1-5, and the consumer companies (vertex set V).

While traditionally, a matching M is the set of graph edges in a bipartite graph in which a vertex is incident to exactly 1 edge in M, for the purposes of this study, a matching M will be defined as the set of graph edges that connect adjacent vertices in a bipartite graph. Depending on the application, a bipartite matching will represent the best pairings of vertices from disjoint sets of vertices.

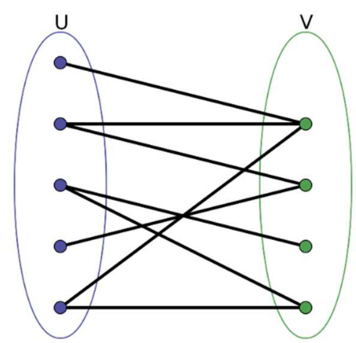


Figure 1-5: An example of a bipartite graph $G = G\ (U \cup V, E)$.

The graph representation for the maximal flow problem is represented by a graph/network, with the vertices representing the starting location (source), destination (sink), and intermediate locations. The graph edges defining the allowable paths from the starting location to the destination. This graph is called a flow network and can be created from a bipartite graph with the addition of a source vertex, sink vertex and the corresponding graph edges between these added vertices and the original vertices of the bipartite graph. This is illustrated in Figure 1-6, showing that one can still use maximal flow algorithms for IS problems once the problem is represented as a bipartite graph.

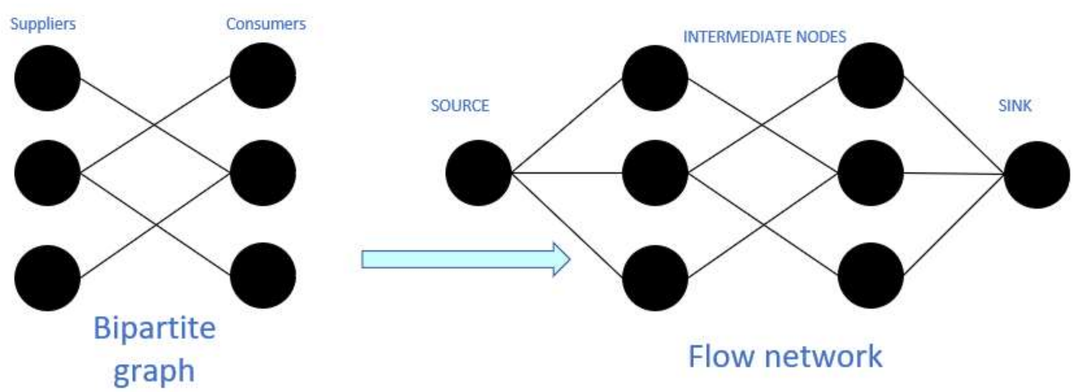


Figure 1-6: Conversion of an IS problem (as a bipartite graph) to a flow network for the application of the maximal flow problem.

A bijection is the matching M for the assignment problem and is, therefore, a subset of the set of graph edges E. So as to differentiate between the types of matchings made from the transportation problem and the assignment problem, it is beneficial to define algorithms that creates a bijection as a *bijective matching algorithm.*

For the creation of an IS network, pairings must be made between suppliers and consumers in a way that is environmentally and economically feasible and should also capture social objectives. This makes the IS problem a multi-objective problem (Yeşilkaya et al., 2020) and usually calls for complex solution strategies to find a globally optimal solution or Pareto-optimal solutions. An instance of this is the creation of an IS network that is optimal with respect to transportation costs and distribution of waste by-product material. While this problem may be solved using linear programs and O.R methods, there are many problems in OR that are NP-Complete. This means that there currently exists no polynomial-time algorithm/method to solve the problem and is noted to be one of the hardest problems in NP (Foulds, 1982). There are few options one can explore when faced with an NP-Complete problem and they are:

1. Develop a methodology to efficiently solve for the optimal solution
2. Find algorithms to solve special cases of the problem
3. Solve for relaxed versions of the problem
4. Devise an algorithm that runs in polynomial time, more often than not
5. Provide heuristics that renders feasible, but not necessarily optimal, solutions

The first approach is unrealistic due to the NP-Completeness of the problem. The second and third case may render results, however, the very fact that the problem is altered means the solution is that of an artificial problem. The fourth case is the best option with the second-best option being the fifth approach (Foulds, 1982). Graph-theoretic modelling allows for the difficult problem to be represented in a way where the structure that results from the modelling approach can be exploited such that at least one of the five approaches listed above may be achieved (Foulds, 1982).

In this way, graph-theoretic modelling allows for robust methods to be explored for the solving of difficult problems. Given the link between IS and the network flow problems, it can be said that the flexibility of graph-theoretic modelling, and the strong graph-theoretic representation of the network flow problems, makes the application of graph theory for the creation of IS networks a promising approach.

2 LITERATURE REVIEW

For more than 50 years, IS has evolved from just a concept to numbering over 100-150 operational or planned IS cases worldwide (Kastner et al., 2015). Shibin et al. (2016) state that ever since the 1990s, sustainable industrial activities have gained international recognition. This drive to sustainable practices in a way that meets social, environmental and financial objectives may be the reason why there is an increase in IS cases around the world (Neves et al., 2020), leading to a large number of case studies produced in the literature (Neves et al., 2020). As a result, numerous works have gone into the development and application of the tools of IS (Lawal et al., 2020), which include PA methods, MP and optimisation, and game theory. On the other hand, much research has gone into compiling, characterising, and deriving the strength and weaknesses of the IS cases (Neves et al., 2020).

IS has also seen research efforts in nontechnical aspects of IS that are focused on trust, communication, and learning which drew scholars to draw knowledge from organisational theories such as social network analysis and environmental strategy (Walls and Paquin, 2015). It is therefore intended to review the available developments in the IS literature and highlight gaps that can be answered by this study.

2.1 Industrial symbiosis Implementation models

IS has become more geographically and institutionally diverse ever since the IS in Kalundborg, Denmark, which played a catalytic role in setting up the field (Graedel and Allenby, 2010). Modern implementation of IS has been in the form of Eco-industrial parks (EIPs), which are industrial parks whose occupants collaborate to minimize energy and waste streams in a way that meets the triple bottom line (Kastner et al., 2015). Chertow and Ehrenfeld (2012) have outlined 5 EIP development models that they identified from IS practices. They are:

- Build and recruit model
- Self-organised Model
- Retrofit Industrial park model
- Planned EIP Model
- Circular economy EIP model

The Build and recruit model is noted to be the most successful (Chertow and Ehrenfeld, 2012) and consists of finding compatible companies to occupy a newly constructed IP (industrial park) so that exchanges may take place. The Self-organised Model explains an EIP in which occupants are privately organised and usually from self-interest. A popular example of this model is the IS in Kalundborg, Denmark. The Retrofit IP model involves converting existing IPs into EIPs, resulting in reduced costs and increased interfirm exchange cooperation (Chertow and Ehrenfeld, 2012). This conversion requires possible connections to be identified from which avenues of constructing the connections are investigated (Kastner et al., 2015).

A Planned EIP model is similar to the Build and recruit model, except that there is a deliberate attempt at identifying compatible companies from different industries, and involves the relocation of the companies into the park so that exchanges may occur. The Circular economy EIP model has been described as a new emerging model in China implemented at the facility level, industrial park level, and regional level and usually incorporates the other models implementing EIPs at the different levels (Chertow and Ehrenfeld, 2012). Chertow and Ehrenfeld (Chertow and Ehrenfeld, 2012) found that Planned EIPs are the least successful model and that the Retrofit IP model's success is dependent on the receptivity from the individual companies on the collaboration implications.

While all five models are based on the underlying principles of IS, much literature has been focused on the Retrofit IP model since there is a great discrepancy between the number of operational EIPs and IPs, with the latter numbering over 12000 (Kastner et al., 2015). Studies have also mostly focused on IS where companies are close to each other in the form of Eco-Industrial Parks (EIPs), with some authors even stating that proximity is a necessary precursor for the success of an IS (Christesen et al., 1999).

While this would ensure transport and infrastructure costs are kept low (Boix et al., 2015) , this view restricts the exploration of synergies between companies that are geographically distant, and hence there is a lack of research into resource distribution at this level. However, there has been some research into IS that stretches beyond the traditional boundaries of IS. This collaboration is called Virtual-Eco-industrial Parks (VEIP). The Brownsville IS in the USA is a typical example of a VEIP, even though it was proposed but never implemented (Zhang et al., 2015). This goes to show VEIPs remain an unexplored possibility with a lack of research into the effects of transport costs and resource distribution over a larger spatial area on the optimisation of Industrial symbioses.

IS brings economic and environmental benefit to the participating companies and key stakeholders, such as the surrounding community (O'Carroll et al., 2014). Benefits such as cost savings, increased resource consumption efficiency and job creation result when transportation and waste treatment facilities are shared (Neves et al., 2019). Also, valorisation of waste material can result in technical and organizational innovations, bringing to light that IS can be a proponent of regional sustainability (Baas, 2011). The extent to which these benefits are achieved is dependent on which connections are formed between the companies. As such, it is the connections that become of interest, ultimately determining the network quality. The goal is, therefore, to find the optimal connections to benefit as many of the occupants without affecting individual profitability.

2.2 Common optimisation methods

Industrial symbiosis incorporates complex trade-offs between the many objectives it seeks to reconcile (Rødseth, 2016). Since IS seeks to benefit many companies (Kastner et al., 2015) a global optimisation is required that results in cooperative benefits in the network. Since it is possible that many companies have their own objectives and constraints, the IS problem can easily become a multi-objective optimisation problem. Environmental and economic objectives are usually antagonistic objectives that multi-objective and many-objective optimisation approaches can be used to satisfy (Cao et al., 2020, Erol and Thöming, 2005).

Hence, to identify feasible IS networks that are optimal for the system (all participants), it becomes imperative to develop and apply optimisation tools to deduce the optimal connections between the participating companies. Without the application of optimisation to the case of IS, IS is just a concept. Much research efforts have gone into optimisation of IS networks. The main optimisation tools are Mixed-integer linear programming (MILP) and Pinch Analysis, which have been used in the processing industry for optimisation at the process-level but were adapted for optimisation among multiple processes and to tackle inter-company exchanges (Kastner et al., 2015).

2.2.1 Pinch Analysis

Pinch analysis has been a central tool in process optimisation and design in the processing industry. It originated from efforts to optimise heat recovery through process integration (Linnhoff and Flower, 1978) as a result of the growing concerns of energy usage in the 1970s (Klemeš et al., 2018). The heat integration is usually in the form of Heat-exchanger networks (HENs), where optimal and thermodynamically possible energy exchanges are created by matching cold streams to hot streams. The term “pinch” refers to the minimum allowable temperature difference between hot and cold streams, which determines the maximum allowable heat that can be transferred. Pinch technology focuses on finding this temperature difference (Rokni, 2016), allowing for optimal hot and cold streams to be matched. The optimal matching can be done using numerical methods, such as the Problem table algorithm or graphical methods following the construction of a cumulative Temperature enthalpy diagram of the streams (Klemeš et al., 2018).

After pinch technology reached a level of maturity, researchers developed water and material optimisation techniques based on the heat pinch technique.

Mass exchange networks (MENs) were first proposed by El-Halwagi and Manousiouthakis (1989) and have been used to treat waste process streams for the removal of pollutants using direct-contact mass transfer units. This is done using mass-separating agents (MSA), which are substances that enable the separation of pollutants from process streams. The optimal allocation of MSAs to pollutant-rich streams can be done using the pinch technique (Isafiade and Short, 2019). Efforts into the synthesis of MENs are either concerned with water streams or non-water-based and gaseous streams, with the former receiving significantly more attention than the latter (Isafiade and Short, 2019). This is due to many mass pinch studies tackling water minimisation problems in water networks since water is a common utility among many industries that face the same constraints such as the shortage of fresh water and discharge permit compliance (Chen and Wang, 2012), leading to the formation of water pinch studies.

The use of the mass pinch technique to specifically address water networks was first done by Wang and Smith (1994) where a water grid diagram was used as a targeting procedure for the development of a water network (Boix et al., 2015).

One of the first applications of water network optimisation at the industrial park level was given in a study by Olesen and Polley (1996) where three localised clusters in the park were optimised individually, after which inter-zone transfers were considered (Kastner et al., 2015). The creation of the water network helps to minimise the consumption of freshwater and the generation of wastewater that would either be treated or disposed. The optimisation by the pinch method is done either by graphical methods or is assisted by other optimisation strategies. Graphical methods are straight forward and easy to interpret when a single contaminant is present in the wastewater, however the method becomes ineffective when dealing with multi-contaminant systems (Boix et al., 2015). Mathematical programming (MP) optimisation is a common optimisation strategy that is used exclusively or coupled with the pinch approach to solve systems that are significantly more complex.

2.2.2 Mathematical programming (MP)

Of the different mathematical programming, Mixed-integer linear programming (MILP) and Non-linear programming (NLP) are most used, especially for water networks (Ramos et al., 2018). Linear programming has been around since the 1950s, a tool developed from Operations research, and has been commonly used for the general problem of allocating limited resources among often competing activities in the most optimal way (Hillier and Lieberman, 2010). There are many situations that are applicable to the scenario of allocating scarce resources among competing activities. Basically, any problem that involves the planning of events/activities to obtain an optimal result can be modelled by a linear program (Hillier and Lieberman, 2010). The application of MP in optimisation studies in the context of Industrial symbiosis is widespread. Studies involving large-scale and complex problems frequently use MP and superstructures (Boix et al., 2015), which are network representations showing the connections and features of the problem.

Takama et al. (1980) used MP to determine the optimal water allocation in a petroleum refinery and showed that a 42% reduction in the annual cost and a 28% reduction in the amount of fresh water used is achievable. Lovelady and El-Halwagi (2009)used a superstructure and NLP for water management in an EIP facility consisting of multiple processes. The authors applied the approach to a case study and showed that the total annualised cost could be reduced to a 10^{th} of the cost when an EIP is not implemented(Lovelady and El-Halwagi, 2009) .

Keckler and Allen (1998) demonstrated that symbiosis in the form of a water network in an IP in Houston, Texas, is possible and economically feasible for all companies in the park and involved the blending of different water streams to obtain various degrees of purity (Keckler and Allen, 1998). The modelling allowed for different scenarios to be evaluated, including a scenario involving the addition of treatment processes, with the authors concluding that the modelling approach can be used for the study of a variety of material flow analyses.

The generation of IS networks based on energy at the park level has not received as much attention in the research community as water networks (Boix et al., 2015). Pinch analysis was adapted by Dhole and Linnhoff (Dhole and Linnhoff, 1993) for the application to an entire plant and is called Total Site Analysis (TSA). Bagajewicz and Rodera (Bagajewicz and Rodera, 2000) developed a TSA approach with a MILP model for heat integration for a site consisting of n plants. The authors pointed out that the geographical position of each plant was fundamental as it affected the pinch temperature and the operating and capital costs (Boix et al., 2015). These types of interdependencies, difficulty in acquiring reliable process data, and the exact resolution needed to be captured by LP models for energy balances make energy networks, in the case of an EIP, quite complex to solve (Boix et al., 2015). This may be the reason why in the literature, energy networks for an EIP are designed but not optimised due to the difficulty in finding an optimal configuration of inter-company exchanges (Boix et al., 2015).

With respect to material sharing in an EIP, there are many instances where by-product material sharing and by-product/waste valorisation are being initiated (Raabe et al., 2017, Collins and Ciesielski, 1994, Nasir et al., 2019, Ram et al., 2020, Haslenda and Jamaludin, 2011, Cimren et al., 2011) with benefits such as a reduction in disposal costs and generation of additional revenue from the sale of by-product materials (Lee, 2012). By-product exchange networks are an effective waste management approach (O'Carroll et al., 2014) that allows for the reuse of by-product materials, leading to a greater reduction in environmental impacts (Heshmati, 2017). However, industrial and solid waste still continues to be a major issue, with municipal solid waste (MSW) expected to grow to 2.2 billion metric tons by 2025 (Vergara and Tchobanoglous, 2012).

The study by Tisserant et al. (2017) found that of the 3.2 gigatons of global solid waste generated in 2007, 1.5 gigatons were sent to landfills. Of the millions of metric tons of plastic that end up in the oceans annually (Gregory, as cited in Vergara and Tchobanoglous, 2012) a 6:1 ratio of plastic to marine debris can be found in some areas (Moore, as cited in Vergara and Tchobanoglous, 2012), illustrating the impact that improper waste management can have on the environment. Hence, while there is a biased focus on increasing energy efficiency among companies in an IS, the generation of waste is outpacing material recycling initiatives (Heshmati, 2017). By-product exchange networks can therefore be used to supplement/reduce the use of landfills as a waste disposal scheme and provide economic benefit to the participating companies while allowing them to meet environmental regulations.

However, there are not many optimisation works that deal with material sharing in an EIP (Boix et al., 2015). It is for this reason that this study is focused on applying graph algorithms for creating matchings for an IS dealing with by-product material exchanges.

2.3 Other optimisation strategies

Optimisation is a process of finding the best solution to a given problem and is used in several fields, making it hard for one to give an exact definition (Weise et al., 2009). However, in many cases, the solution strategy incorporates optimisation algorithms, which are guided by mathematical functions (Weise et al., 2009). These functions collectively generate a solution space, which can be thought of as a multi-dimensional region that may consist of multiple maxima and minima and contain areas that are not continuous (Weise et al., 2009). Solution space topologies of this nature make solving the problem particularly difficult. In other words, finding a solution that is globally optimal becomes a tedious task, especially in the case where multiple local optima exist (Weise et al., 2009). In such cases, it becomes difficult to determine if the incumbent solution is an optimal over the whole solution space or just in its locality. As such, much research has been devoted to developing different optimisation approaches for tackling different optimisation problems with this type of nuance (Weise et al., 2009).

Of the different approaches to optimisation, Mathematical programming is but one approach. There are other approaches, such as the use of greedy algorithms, dynamic programming, systematic search, and metaheuristics (Mehlhorn and Sanders, 2007). While greedy algorithms can produce solutions fast, with some greedy algorithms able to find the optimal solution (such as minimum spanning tree algorithms), the majority of greedy algorithms usually provide low-quality solutions because the algorithm gets trapped in a local optimum in the solution space (Mehlhorn and Sanders, 2007).

Dynamic programming consists of creating an exhaustive list of optimal solutions, in which the preceding solutions are for subproblems of smaller size to the actual problem and are used to construct the solution for the larger problem. This could require significant computational space and time, with the problem needing to be of a special structure (Mehlhorn and Sanders, 2007). Systematic search techniques are a brute force approach that seeks to enumerate through the possible solutions to find the optimal one. A common approach to the systematic search is the branch and bound approach, in which the

feasible set of solutions to the problem is represented as a tree with smaller subsets of solutions partitioned as branches of the tree. The search then explores the branches of the tree, with a bounding rule that stops the search on certain branches if it is determined a more optimal solution than the incumbent solution cannot exist on those branches (Mehlhorn and Sanders, 2007). However, such techniques are slow and may be infeasible when the search space is large (numerous possible feasible solutions) (Hajebi et al., 2015).

Metaheuristics are a commonly used procedure in the field of operations research and combinatorial optimisation (Gendreau and Potvin, 2005). As opposed to heuristics, which are specialised for a particular problem, metaheuristics are a general solution scheme that can be adapted to be used to solve different types of problems (Gendreau and Potvin, 2005). Gendreau and Potvin (2005) presented an overview of metaheuristics and its applications in the field of combinatorial optimisation. The authors found that there are two distinctions of metaheuristics, Single Solution Metaheuristics (SSM) and Population Metaheurisitcs (PMH). SSM considers a single solution at a time (Gendreau and Potvin, 2005) and is usually based on local search techniques, with common examples being Simulated Annealing, GRASP, Tabu Search, and Variable neighbourhood search. PMH is different in that a multiplicity of solutions are explored concurrently (Gendreau and Potvin, 2005), with common examples being the Ant Colony optimisation and the Genetic Algorithm (Gendreau and Potvin, 2005).

Not much research has gone into applying these other optimisation methods for the case of industrial symbiosis, with only a few works being done using PMH methods (Zheng et al., 2013, Ren et al., 2016, Hajebi et al., 2014). Hajebi et al. (2014) have noted that metaheuristics do not consider the specificities of the problem, such as the structural requirements of the network. For instance, the structural requirements of a water distribution network (WDN), such as network partitions and connectedness, have been identified by Hajebi et al.(2014) as important considerations in the design of the WDN. In fact, many studies often do not consider the structural features of the network when designing and optimising IS networks (Doménech and Davies, 2009). Excess network complexity, due to not considering the structural features of the network, is a problem since, for the case of interplant connections where companies are separated by relatively large distances, the network complexity can be considered as an investment cost (Boix et al., 2015) and therefore factors into the feasibility of the network/IS connections (Boix et al., 2015). Boix et al.(2012) found that factoring in the number of connections into the objective function for the case of waste water management between industrial companies allowed for the reduction in freshwater consumption and led to an increase in efficient inter-plant exchanges. This demonstrates the importance of considering structural features of the resulting IS network and can assist in choosing the best alternative given multiple feasible networks for the same problem.

2.4 Decision-support tools and commercial software

While an EIP can be optimised in many ways, the task of optimising one with respect to profitability, environmental performance, and network operability is a difficult one as it has been noticed that numerous research works often deal with systems that have conflicting objectives (Boix et al., 2015) and this is especially true for energy networks. As a result, many authors often simplify the problem to have an LP that is easier to solve (Boix et al., 2015). Integration efforts at the inter-company level have been made using multi-periodic MILP models (Fichtner et al., 2004, Hirata et al., 2004, Kim et al., 2010) of the energy and material flows of the firms with the aid of a mathematical programming language called GAMS (Brooke et al., 1996).

Karlsson (2011) created a decision support tool called MIND (Method for analysis of INDustrial energy systems), which is a software developed in Fortran and based on MILP. Karlsson and Wolf (2008) implemented this tool to demonstrate industrial symbiosis in a forest industry case study and showed that the integration of the companies based on the material and energy connections determined by the decision support tool brings economic benefit to the companies compared to operating alone. Cimren et al. (2011) used the Eco-Flow™ software to show the economic and environmental benefits of a by-product exchange network in an IS network in Missouri. While these approaches are sophisticated and effective as decision support tools, they can't find an optimal solution, rather they enable the identification of many feasible solutions.

To this end, many studies conclude their optimisation with post-optimality decision strategies to evaluate the best alternative among the solutions rendered by the models (Boix et al., 2015).

2.5 Post-optimality decision support tools

Multi-Criteria Decision tools, such as the Weighted Sum Method (WSM), Analytical Hierarchy Process (AHP), and the Technique for Order Preference by Similarity to Ideal Solutions (TOPSIS), are used in multi-objective optimisation problems and allow decision-makers to choose the best solution based on their preference. The generation of the solution alternatives is typically done by Pareto optimisation for multi-objective optimisation studies (Limleamthong and Guillén-Gosálbez, 2017). Pareto optimisation is a solution to multi-objective optimisation studies that can be used to identify the set of points that lie on the boundary of the feasible criterion space, known as the Pareto-optimal points (Athan and Papalambros, 1996). The points represent solutions that have the property of not being able to be improved in one objective without seeing a worse performance in another objective. The non-inferiority of the solutions with respect to each other means that decision-makers can choose the Pareto-optimal alternatives that best reflects their preferences. This may be interpreted as the choice of distribution policy, since it is a policy/ optimality criterion that determines the distribution of the waste flows in the IS network.

In other words, the choice of alternative is usually dictated by the policy the decision-makers most value, whether it is environmental, social, or economic. Furthermore, IS studies assess at least one of these benefits (Neves et al., 2019). While the economic objectives are most easily mathematically formulated (Boix et al., 2015), usually by way of an economic indicator (such as the net present value) or a single composite objective function (Boix et al., 2015, Madani, 2010), social and environmental objectives are less so.

2.5.1 LCA: Environmental objectives

Life Cycle Analysis (LCA) is a formalised method (ISO 14040:2006;ISO 14044:2006) for the identification and quantification of material flows at every step of the process and is widely used in the literature to assess the environmental impacts/benefits of IS networks (Neves et al., 2019). Its procedures are recorded in the 14000 series of ISO. It has also been used in multi-objective optimisation studies as a post-optimality decision tool for Pareto-optimum solutions to identify more sustainable options (Azapagic, 1999, Boix et al., 2015). Ardente et al. (2010) conducted a study in which the authors applied LCA to determine the most environmentally friendly IS configuration between companies in Southern Italy that were identified as a possible Polypropylene (PP) based IS network. The study showed that the use of LCA allows for the management of complex information that is required for identifying connections. The LCA was applied to 3 economically feasible scenarios that involved the production of car upholstery from PP. The authors found that the use of 90% recycled PP, obtained from the companies in the network, to produce car upholstery, was more environmentally friendly than using 100% virgin PP. A reduction in the total energy consumption, waste generation and raw material consumption was found (Ardente et al., 2010).

Hence, the use of LCA is a viable method for the identification of environmentally friendly IS networks. However, it is a very data-intensive method (Finnveden et al., 2009), meaning that it is a costly and lengthy process. LCA also lacks the methodologies to assess water consumption (Friedrich and Buckley, 2002), and the flexibility of the ISO rules leads to concerns of the reliability and consistency of findings from different LCA studies on the same system (Gava et al., 2018). LCA is also limited to environmental aspects, usually omitting the economic and societal aspects of the symbiosis (Gava et al., 2018).

Since the implementation of IS involves decision-makers cooperating and engaging in social relations, the societal impact of IS networks is an important factor as it speaks to the degree to which companies are willing to participate. A key facilitator to the progression of IS networks is a high level of trust between the companies involved (Hewes and Lyons, 2008). When this is lacking, company participation and network stability in the long term come into question. Game theory is well suited to these types of problems and has been used in the literature as a decision-making tool to facilitate cooperation in IS given different IS solutions. (Palafox-Alcantar et al., 2020).

2.5.2 Game theory: Economic objectives

Game theory is a set of mathematical tools that can be used to analyse the interaction among decision-makers (Palafox-Alcantar et al., 2020) given some strategic interaction. It assumes that a decisionmaker/player is rational and intelligent (Chew et al., 2009) and makes decisions that render the best strategic advantage for itself in a game/strategic interaction. The game is classified as non-cooperative if players don't agree to coordinate their strategies (Chew et al., 2009) and cooperative if players do. In a non-cooperative game, the Nash equilibrium is the solution from which any unilateral change in decision from any single player will not improve his/her payoff (Chew et al., 2009), whereas, in a cooperative game, a Pareto-optimal solution can be reached between the players.

Game theory determines the most probable outcome based on the assumption that players make self-interested decisions. It has mostly been used for the assessment of the stability of IS networks and the analysis of players' behaviours over time (Jato-Espino and Ruiz-Puente, 2021), using computational modelling tools such as Agent-based modelling.

IS solutions generated by conventional optimisation methods assume perfect cooperation, when in reality companies optimise their own objective rather than optimising the IS system (Madani, 2010). This tends to lead to the collapse of the IS negotiations and is referred to by Yazdanpanah et al. (2020) as the *Industrial symbiosis Implementation* problem. Consequentially, the majority of the research focused on the application of game theory to IS networks is in the distribution of costs and benefits among the participants via cooperative game theory(Yazan et al., 2020, Madani, 2010) using allocation methods. It has also been used for analysing the stability of IS networks, rendered by company strategies (Yazan et al., 2020, Palafox-Alcantar et al., 2020).

Andiappan et al. (2016) used a cooperative game theory approach to show that annual savings in the range of 14% to 48% is achieved when the three companies involved adopted a cooperative attitude. Parlar et al. (2019) used cooperative game theory to show that an increase in the NPV of 3 hypothetical companies is possible if the companies agree to share the benefits of the symbiosis. Chew et al. (2009) applied game theory to model the interactions of participants in an EIP and found that a higher collective payoff was achievable when a cooperative scheme was used as opposed to the non-cooperative scheme. The work done by Grimes-Casey et al. (2007) and Yazdanpanah and Yazan (2018), who have applied cooperative game theory to IS, concluded that cooperative game theory could be used to render fair and stable IS networks. However, Grimes-Casey et al. (2007) did state that government regulations/incentives are required to support cooperative game theory solution concepts.

2.5.3 Social network analysis: Social objectives

Another method that has been used to study the social mechanisms of IS is Social Network Analysis (SNA). SNA is a research method that uses graph theory and network analysis to formulate quantitative measures for the analysis of qualitative concepts (Adhikari, 1960). It has been used in IS to analyse the structural attributes and internal functional characteristics of IS networks (Zhang et al., 2016) using metrics quantified from the network abstraction of the IS.

Domenech and Davies (2011) used SNA to understand the social aspects of IS and analyse the interaction of the participants. The authors used three different measures of centrality in the hope to understand the structural attributes of the Kalundborg EIP that make it so successful and found that the occurrence of a network core and periphery area allowed for ease of interaction between the main members of the IS network, with the former being dense and well-articulated (Domenech and Davies, 2011). The average distance between nodes was found to be small and meant that transportation costs were low. The analysis also enabled the identification of 3 key players in the network, with the risk of network fragmentation and network instability if these companies are removed from the IS (Domenech and Davies, 2011).

Song et al. (2018) analysed the Gujiao mining IP in China using SNA and found that while the network seemed to be well connected, waste exchanges could be increased since the metric value to measure network density was low. Zhang et al. (2016) used two metrics to compare the degree of completeness and dependence of 8 different IS networks, which allowed for the categorisation of the IS network into 3 distinct types. This categorisation allowed for the authors to comment on the degree of instability that could result under different threats (such as the departure of key participants) (Zhang et al., 2016). The authors were also able to identify new connections that would improve the network density, resulting in increased water usage efficiency (Zhang et al., 2016).

The benefits of SNA are in its abstraction of the participants in an IS to a graph, with a focus on the relational information of the participants (Doménech and Davies, 2009). This allows for the identification and proposition of structural network features that lead to the success of IS. These structural features are the graph metrics of the graph, which have been instantiated from the field of network analysis and has been used in SNA to allow for descriptive analysis of IS networks and characterisation (Song et al., 2018, Zhang et al., 2013, Chopra and Khanna, 2014). However, the implementation of SNA is mostly focused on already existing IS networks with suggestions on network improvement based on connections not being as automated and deterministic as the structural analysis it affords.

Since the interest is in matching suppliers and consumers in an optimal way, which in turn generates the connections in the IS network, the IS problem may be considered as a matching problem. Matching problems, to which matching algorithms are applied, are found in many fields of research, such as automatic data processing, image processing, and artificial intelligence (Tanimoto et al., 1978). In lieu of its importance to matching problems, a review of matching algorithms is necessary.

2.6 Matching algorithms

At a high level, matching can be considered as the general case of mapping set(s) of objects to each other. Framed in this way, matching problems can consist of finding matchings between one, two, or three sets of objects (Fàbregas Vázquez, 2015). Matching problems between two sets of objects (known as Two-sided Matching) will be focused on as it has direct relevance in the context of IS.

Two-sided Matching was introduced by Gale and Shapley (1962) and has to do with matching one kind of object to another. It seems that from the literature, the application of matching algorithms has mostly been on determining the equivalence of objects or for determining the optimal assignment of one set of objects to another, both of which can be considered one and the same at a higher level.

Arguably the most common application of matching algorithms is in string matching, which is one of the most studied problems in Computer science (Monge, 2000). It has a wide range of applications such as bioinformatics, plagiarism detection, information security, and text matching (Al-Khamaiseh and ALShagarin, 2014). The main approach used in string matching is based on edit distance, which is the minimum number of operations on individual string characters to transform one string of symbols to another (Monge, 2000).

Graph matching

Technologies like pattern recognition and computer vision also depend on identifying object similarity to work (Bunke, 2000), both of are linked to the problem of correspondence, which is concerned with finding the mapping between one set of points to another (Cour et al., 2007). This is usually enabled after the graph transformation of an image's/ video content's feature points as graph vertices and edges (Bunke, 2000), transforming the problem to a case of graph matching (Cour et al., 2007).

Graph matching has been used for the past 30 years for pattern recognition as well as for other applications where the structural information of the objects to be matched are represented as graphs, such as in chemical structure analysis (Rouvray and Balaban, 1979), one of the earliest implementations of graph matching (Bunke, 2000). Bunke (2000) states that other applications of graph matching are in machine learning, monitoring computer networks, and determining the similarity of conceptual graphs.

The schema matching problem is another important problem in data application scenarios where semantic heterogeneity is a severe issue (Algergawy et al., 2007), such as preventing the seamless transfer of information between heterogenous sources in electronic business transactions (Kim et al., 2011). It is a problem that seeks to find the semantic relationships between different data repositories/ elements of schemas (De Carvalho et al., 2013). The identification of structural and semantic similarities is enabled by the conversion of the schemas to a tree representation (acyclic graph with a root node) (Kim et al., 2011).

The widespread application of graph theory for the case of matching is appropriate since, in graph-theoretic terms, a matching is a graph $G = (V, E)$ whose set of edges E don't share a common vertex in V (Tarjan, 1983). In other words, every node is incident to at most one edge in E. Therefore, matching, in the case of graph theory, induces the pairing of the nodes in graph G (Ahuja et al., 1995). The graph edges may have a certain numerical value/weight associated with them, which can be interpreted as the cost of the pairing between the two nodes the edge connects. Matching can be split into two general categories:

1. Bipartite matching
2. Non-bipartite matching

Bipartite matching refers to a matching on a bipartite graph, which is a graph $G = (V, E)$ whose vertex set V can be divided into two disjoint sets of vertices, with an edge from E connecting a vertex from one set to the vertex of the other (Skiena, 1990). On the other hand, Non-bipartite matching refers to matching on graphs that are, by definition, not bipartite graphs and are termed as non-bipartite graphs. Non-bipartite matching is generally a more complicated task than bipartite matching (Tarjan, 1983). Ahuja et.al. (1995) states that matching problems can be further classified as either cardinality matching problems or weighted matching problems.

While the cardinality matching problem deals with maximising the number of nodes matched, the weighted matching problem is focused on minimising the weight of the matching, where the weight of a matching is the sum of the edge weights for those graph edges in the matching (Ahuja et al., 1995). Hence, given a bipartite graph, the weighted matching problem seeks to find the matching between the two disjoint sets of vertices that is of minimum weight/ cost. This weighted matching problem on a bipartite graph is particularly relevant to the case of IS, as the two disjoint sets of vertices of a bipartite graph, G, can represent the two sets of companies, suppliers and consumers, with the edge set E of the graph representing the connections that exist between the companies, forming an IS network. More importantly, since the optimality of the IS network are sensitive to which supplier and consumer pairs are deduced, finding the matching of minimum weight allows for the identification of the optimal pairing between the companies in the Industrial symbiosis.

There are many graph algorithms that can be used to create matchings on bipartite graphs, particularly network flow algorithms that are designed to solve network flow problems. The network flow problem is a broad classification of problems that are concerned with minimisation of the cost (usually transportation costs) of moving materials through a network from the initial location of the materials to the locations where the demand for the materials exists (Vanderbei, 2020). Clearly, the application to IS becomes relevant, especially for the case of material sharing in an EIP, which has not received as much focus in the optimisation works as the other forms of EIP integration (such as water and energy) (Boix et al., 2015).

A common network flow algorithm is the Ford Fulkerson method (Ford and Fulkerson, 1956) for maximal flow problems, which is a fundamental problem in network flow theory (Kyi and Naing, 2018). The Ford-Fulkerson method encompasses several implementations (Cormen et al., 2009), such as the

Edmonds-Karp algorithm, and has been used in various network-based applications (Neto and Callou, 2015, Abdullah and Hua, 2017, Kyi and Naing, 2018). The Hungarian method is an algorithm, developed by H.W Kuhn (Kuhn, 1955) designed for the assignment problem, which is a weighted-bipartite matching problem that seeks to find the optimal matching between two distinct and independent sets of objects such that the cost of the matching is minimised (Ahuja et al., 1995). It has been used in many assignment and matching applications, such as DNA computing (Frank, as cited in Zeng et al., 2014), Graph matching (Kriege et al., 2019, Fröhlich et al., 2005), and clustering applications (Goldberger and Tassa, 2008).

The network simplex method by George Dantzig (Dantzig, 2016) is also an applicable algorithm for the weighted matching problem. It was the first algorithm designed for network flow problems and is a highly streamlined version of the Simplex method (Hillier and Lieberman, 2010) that takes advantage of the combinatorial nature of network flow problems (Goldberg et al., 1989). This study, however, makes use of the Primal Simplex method as it is not only extremely efficient in practice (Ahuja et al., 1999) but also offers the opportunity of comparing the graph-based algorithms used in this study (HM and FF) over a linear-algebraic method like the 2-Phase Primal simplex method (SM).

While there are instances in the literature where graph representation of IS systems are used (Hein et al., 2015, Aviso et al., 2015), these are mostly from SNA studies on already existing IS scenarios and used for descriptive analysis or make use of MP approaches for obtaining optimal solutions. Graph theory has mostly been used to model systems, from which MP methods are used to optimise the system. One wonders why the use of graph representations is not met with the complimentary use of graph algorithms in many studies. With respect to the use of graph algorithms, the most common application was found to be graph decomposition in WDNs or for the identification of spanning trees/critical paths in the WDNs (Zheng et al., 2013, Kessler et al., 1990, Kadu et al., 2008). To the best of the author's knowledge, there hasn't been any research that has used graph-theoretic representation to model IS with graph algorithms to deduce IS connections. More specifically, the question arises as to why the use of fundamental O.R problems that have strong graph representations has not been used to model the IS problem?

2.7 Summary

While there is active development in the field of IS, there are concerns regarding the practical implementation of IS, specifically surrounding the fairness and stability of the network (Yazdanpanah et al., 2020), which can be linked to structural features of the IS network. Furthermore, popular models of IS, i.e., EIPs, are mostly limited to situations in which close geographical proximity is a necessary precursor and, as such, involve companies within an IP. There are not as many VEIP studies done in which geographical proximity is not a necessity. While there are many optimisation tools available for IS network optimisation, many are either limited or are required to be inherently complex. Solutions from such complex solution strategies usually produce Pareto-optimal solutions that are optimal with respect to different distribution policies.

Furthermore, optimisation works on by-product exchange networks are not often conducted in the literature and are usually biased towards energy synergies between companies. It has been found that MP and Pinch methods are the dominant optimisation methods used by researchers for IS even though these methods have drawbacks. While there are many graph-based studies, most are SNA studies focused on gaining descriptive and prescriptive analysis of already implemented IS. As a result, not many graph-based techniques have been applied to IS as optimisation tools. Structural features are usually overlooked, especially when local search methods and metaheuristic approaches are used. This has been highlighted as one of the causes of excess network complexity in IS network solutions. It is therefore found that there exist significant gaps in the literature, which is deemed important to fill for the progression of the field.

3 THESIS OBJECTIVES

Based on the review of the literature in Chapter 2, it was noted that while there are significant efforts in optimisation methods for the design and creation of IS networks, many research efforts don't incorporate multiple objectives in the optimisation process (Boix et al., 2015). This results in the multi-objective nature of the problem posing severe challenges to the possibility of finding a global optimum. Rather, Pareto-optimal solutions are generated for multi-objective studies (Limleamthong and Guillén-Gosálbez, 2017) from which a solution is chosen that best reflects the distribution policy favoured by the decision-makers. While material sharing IS networks have been noted to be more of an identification problem than an optimisation problem (Kastner et al., 2015), there are not as many research optimisation efforts regarding this type of exchange (Boix et al., 2015), specifically dealing with by-product and waste exchanges in which transportation costs are a contributing factor.

Optimisation works often don't consider structural features of an IS network when trying to deduce optimal connections in IS networks (Doménech and Davies, 2009), which can lead to excess network complexity and result in long-term instability (Hajebi et al., 2014). It was noted that most studies of a graph-theoretic nature involve the application of SNA, offering more of a descriptive analysis on already existing IS networks than in deducing IS networks. Given these challenges, the potential is seen for a graph-based approach for creating IS networks. Furthermore, by making the study an instance of an IS concerning the exchange of a single waste by-product material, a contribution towards optimisation for material sharing IS networks will be made.

A graph-theoretic representation of the IS problem may ameliorate the difficulties in capturing multiple objectives by capturing a compact representation of the problem. Three O.R problems have been identified as suitable candidates to model the IS problem, viz., the transportation problem, optimal assignment problem, and the maximal flow problem, all of which have strong graph representations.

These O.R problems are related to well-established graph algorithms that can be adapted for the case of matching in IS. For the optimal assignment problem, the application of the Hungarian method (HM) offers a graph-based approach of obtaining optimal pairings between two distinct sets of objects, such as suppliers and consumers. For the maximal flow problem, the Edmonds-Karp algorithm (FF) can be used to determine the flow network that maximises the flow of a resource from a set of starting locations (suppliers) to a set of destinations (consumers). Lastly, for the transportation problem, the 2-Phase Simplex method can be used to find the solution to the IS problem, defined by a linear program. The application of the Simplex method also provides the opportunity to determine if any distinction can be made between linear-algebraic approaches and graph-theoretic approaches (FF and HM).

In this way, the study will make use of graph-theoretic modelling and the application of graph algorithms to deduce optimal IS networks between companies that are drawn together by the exchange of a single waste by-product material. This thematic graph-based approach naturally allows for the opportunity to extract graph metrics that will allow for the structural characteristics to be evaluated as a decision support tool. Hence, structural features of the IS network can be inherently facilitated in the approach, which is beneficial given that SNA has been identified as an organisational theory to study participant trust in IS networks (Walls and Paquin, 2015).

Since the three algorithms belong to 3 different classes of network flow problems, there is a possibility of different IS networks being rendered. The graph-based approach will, therefore, make it possible to identify graph characteristics unique to the algorithms and how it translates to distribution policies of an economic, social, and environmental bent.

Hypothesis

The structural features of an IS network can be used to evaluate the effect of distribution policies on IS networks created by graph matching algorithms when they are adapted to the IS problem.

The aim of this research is to investigate the suitability of a few fundamental algorithms for the creation of IS networks that are feasible with respect to transportation costs and resource distribution under different distribution policies. Of specific interest is determining which of the algorithms is superior for different distribution policies

To achieve this, the following objectives need to be met:

- The development of a graph-theoretic model to represent a set of industrial companies, including all the necessary meta-data required to explore a suitable graph representation.
- The development of a method that makes it possible for feasible IS networks to be rendered that satisfy all imposed environmental and economic constraints using fundamental Operations Research algorithms and graph theory algorithms.
- The development of a framework to facilitate the analysis of the graphs rendered by each algorithm for determining the suitability of algorithms under different distribution policies.
- Application of the algorithms to an IS case study to quantify the benefits derived from the IS network created by each algorithm.

In the notation of Lütje and Wohlgemuth (2020), this study is focused on the facilitated synergy identification. The study covers the use of graph-based matching algorithms for the identification of IS opportunities that specifically deals with by-product/waste synergies with the aim of reducing transportation costs and achieving resource distribution. The first step in this endeavour is to first define the problem formulation and ensure that a suitable graph-theoretic model can be developed from it. Such a model will provide the foundation for graph representations to be created for each algorithm. To investigate the suitability of the algorithms in creating IS networks, a case study would have to be solved.

A deeper understanding of the algorithms must first be gained. A rigorous study is conducted, that follows a pre-defined framework, by creating machine-generated datasets consisting of the participating companies' information relevant for creating matchings. This allows for investigations into whether the graph algorithms render unique graph characteristics in the IS networks created. From this understanding, the suitability and ranking of the algorithms under different distribution policies can be inferred and tested. Accordingly, the next two chapters are focused on the representation of IS networks and the development of a framework and methodology to investigate whether the algorithms afford beneficial graph characteristics to IS networks.

4 MODELLING IS NETWORKS AS GRAPHS

In this chapter, the problem formulation is specified and followed by the development of the Graph Theory-based modelling approach. However, the motivation for the formation of these graphs/networks is reviewed first. The IS matching that renders itself as an IS network is based on supplier-consumer pairings. These connections should respect a distribution policy that that will promote and encourage IS membership activity while still meeting the overall objective of reducing virgin material consumption.

4.1 Distribution policy

Matching in an IS network is subject to several, possibly contradictory, objectives. In addition to the original stated objective of reducing raw material consumption, it is only intended to promote the profitability of the participating member companies, and still maintain equity of distribution among the partners. In other words, the anticipated profitability increases should be as evenly distributed as possible. Therefore, the following categories of policies are identified.

These policies may have an Environmentally biased Distribution (EVD), Economically biased Distribution (ECD), or a distribution policy based on Social Construction (SCD). An ECD policy is one in which the aim is solely focused on cost savings. An EVD policy is a policy that prioritizes a reduction in environmental impact rather than economic gain. An SCD policy is when the emergence of social constructs like trust, network stability, and competitive integrity is paramount in the IS network.

In an SCD policy, companies attempt the creation of a matching that prioritises trust among companies, or one in which as many companies as possible gain competitive advantage. According to Capitalist theory, this is considered a fair distribution. This can be linked to cooperative behaviour from Game Theory. On the other hand, companies may seek to gain dominance, following non-cooperative behaviour. As a result, hierarchies are formed with a distribution that more beneficial to some groups in the collaboration (e.g., the suppliers) compared to others (consumers).

While one can't deny that an ECD policy may bring a reduction in environmental impact or that an EVD policy may have some measure of stability, the distinction in distribution policy outlines that companies favour a specific outcome without being much concerned about optimising the benefits the other distribution policies bring.

The implications of the distribution policies are listed below:

1. Minimising transportation costs (ECD)
2. Minimising waste material in landfills/water bodies from by-product disposal (EVD)
3. Minimisation of resource purification costs (ECD)
4. Minimising the power imbalance between supplier-consumer pairings (SCD)

It is seen that point 1 may also result in the minimisation of greenhouse gas emissions and point 2 will result in the reduction in waste disposal costs. However, proportionalities like this may not be as evident in other cases. Point 1 would create matchings between companies that are in close geographical proximity, while point 2 would seek supplier-consumer pairings that result in the smallest amount of by-product material, irrespective of transportation costs. Point 4 would distribute by-product material such that a similar benefit is derived for all pair-wise relations.

With the understanding that it is the distribution policies favoured by stakeholders that drive the IS collaboration, the general IS formulation may be given.

4.2 IS Problem Formulation

There are a group of companies that have a common interest in the exchange of a single by-product material and can be separated into two subgroups viz. a set S of suppliers ($|S| = m$) and a set C of consumers ($|C| = n$). The problem is to create an IS network by matching the set of suppliers and consumers such that the policies that the companies collectively favour are upheld. A supplier-consumer pairing is stronger if it upholds the specified policies to a greater extent.

Each company has provided a set of data, which will be called the *product attributes*, that that are required to deduce supplier-consumer pairings. The product attributes from the companies are used to quantifiably represent the policies valued in the collaboration. For instance, product attributes could be comprised of the geo-location of the company, amount of by-product material required/available, the economic value of the by-product (selling/buying price), and the product purity. In this way, the transportation costs, purification costs, and resulting by-product material as landfill waste can be calculated based on which supplier-consumer pairings are created.

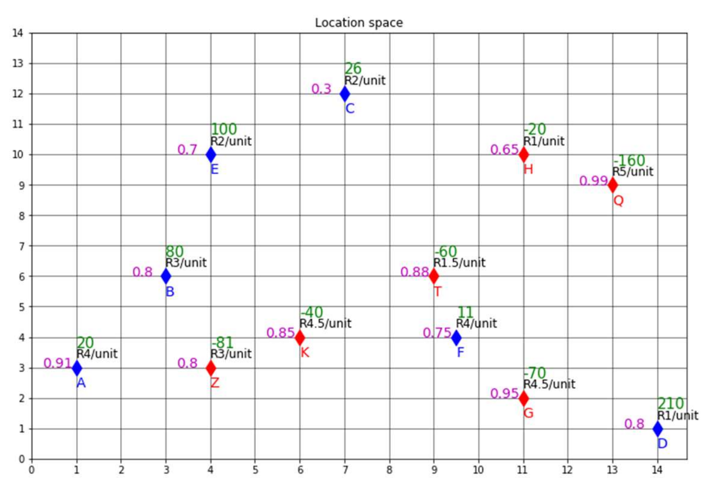


Figure 4-1: Generalized problem formulation.

Figure 4-1 illustrates the problem formulation in geographical space. The blue dots represent the set of m suppliers ($m = 6$) and the red dots represents the set of n consumers ($n = 6$). The coordinate space represents the geographical location of a company, with company data shown at the geo-location for each company. The values in magenta represent the percentage purity of the by-product material and this would be the purity available at the supplier and the purity desired by the consumer. The unit price proposed by each company for the by-product is shown in black, and the values in green represent the amount/capacity of by-product material, where:

$$Resource\ Capacity\ of\ Company\ i = \begin{cases} x_i & if\ \ i \in S \\ y_i\ if & i \in C, y_i < 0 \end{cases} \qquad \text{Equation 4-1}$$

4.3 Graph modelling of the problem formulation

The graph modelling entails representing the problem formulation as a graph. The problem formulation stipulates that the focus is to create a feasible IS network with respect to the policies. These policies impose constraints on the problem and take into consideration the companies' attributes when creating an IS matching through supplier-consumer pairings. With the companies represented as graph vertices, a bipartite graph becomes the suitable graph-analogue, since there would be 2-distinct and disjoint sets of vertices, with one set of vertices being of size $|m|$ (m suppliers) and the other of size $|n|$ (n consumers).

The pairing of a supplier to a consumer can be represented by an edge e whose weight w is a function of the constraints, and it reflects the strength of the pairing. In this way, all pairings between the two sets of vertices in the bipartite graph make up a matching M. Figure 4-2 illustrates a bipartite graph. The set of edges formed between the two sets of vertices/companies is the matching M, with every edge between a company pair (S_i, C_j) representing the exchange of monetary goods and by-product material. For example, the edge e_{11} represents a pairing between supplier one and consumer one where the supplier would provide x_1 units of by-product material to the consumer that required y_1 units of by-product material.

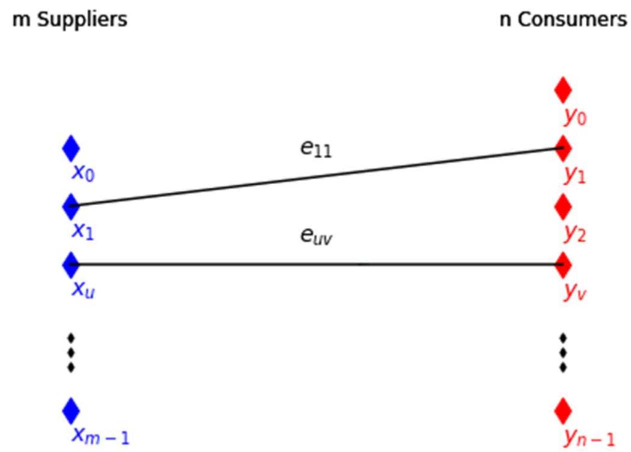


Figure 4-2: A generalized illustration of a bipartite graph G, showing two matchings.

Since the weight of an edge reflects the strength of the pairing, the edge weight becomes a function of each company's product attributes for the two companies (supplier and consumer).

The company's product attributes (PA) are listed below:

1. σ = Resource capacity
2. α = Unit price for the resource
3. φ = Purity of the resource
4. θ = Geographic location of the resource

Since the ideal matching M would be the subset of edges that create strong supplier-consumer pairings, one first needs to deduce the edge weights for all possible supplier-consumer pairings. An adjacency matrix can be used to represent all possible supplier consumer pairings and is usually used to represent a graph $G = (V, E, W)$, where W represents the weight/strength of edges/pairing.

The adjacency matrix $\underline{A}$ can be computed using Equation 4-3 by first computing an assignment matrix $\underline{\acute{A}}$ using Equation 4-2 .It becomes relevant in later sections and is therefore beneficial to derive. For the case of IS, the graph G is the bipartite graph whose set of vertices $V = S \cup C$ are a made up of 2 disjoint subsets of vertices S and C. The assignment matrix $\underline{\acute{A}}$ is one that can be used to show the pair wise relations between a supplier and consumer, with the element $\acute{A}_{ij}$ quantifying the strength of a relationship between supplier i and consumer j according to some cost function $f(PA_{S_i}, PA_{C_j})$ that takes into account the product attributes for supplier i $\left(PA_{S_i}\right)$ and consumer j (PA_{C_j}).

$$\underline{\acute{A}}: \acute{A}_{ij} = f(PA_{S_i}, PA_{C_j})$$

In practice, the generation of the matrix $\underline{\acute{A}}$ is done using $2 \times d$ sets of lists, where d is the number of product attributes to weight the edges and there are two sets of these lists, one belonging to the suppliers and the other set to the consumers. Hence, if the company resource capacity σ was the only parameter to weight the edges then the σ list for the suppliers and consumers would be the input to the cost function for quantifying pair-wise supplier-consumer connections:

$$\underline{\acute{A}}: \acute{A}_{ij} = f\left(\sigma_{S_i}, \sigma_{C_j}\right), i \in [1, m], j \in [1, n]$$

$$\sigma_S = [x_1, x_2, x_3 \ldots x_m]$$

$$\sigma_C = [y_1, y_2, y_3 \ldots y_n]$$

$$\underline{\acute{A}}\ is\ a\ p \times p\ matrix, p = n\ if\ n \geq m, else\ p = m$$

Equation 4-2

$$\underline{A}[0, p{:}\,p + p] = \underline{\acute{A}}$$

$$\underline{A}[p + p, 0{:}\,p] = transpose(\underline{\acute{A}})$$

Equation 4-3

5 MATCHING INDUSTRIES BASED ON DISTRIBUTION POLICIES

Regulation improves on the environmental performance of the chemical industry, and this is required in the chemical industry due to its negative reputation in the eyes of society with regards to accidents and toxic emissions that affect the lives of the surrounding community and the environment (National Research Council, 1999). The depletion of natural resources and waste disposal are also areas in which the public associates with the chemical industry (Clarck, 2003). With the increasing load of waste material on landfill sites and the contribution of wastewater effluents on water pollution, solutions are required to minimize waste disposal. Industrial symbiosis aids in this regard as not only does the distribution of by-product material among companies aid in the reduction of by-product material disposal and pollution, but it also reduces the amount of virgin raw material consumed by the companies. This results in increased resource efficiency and lower environmental impact.

This chapter is focused on the application of the graph algorithms to IS, with specific emphasis on minimising by-product material that ends up as waste while minimising transportation costs. Hence, the distribution policy is mainly an EVD and ECD type policy. This requires the graph representation to be established for each network flow problem that the graph algorithms are suited for.

The chapter is split into subsections, with each subsection dedicated to a specific network flow problem and its corresponding algorithm. For each subsection, a brief introduction of the network problem will be provided before the graph representation is presented. This will then be followed by the required adaption of the graph model to the IS problem and the solution strategy of the algorithm.

It will be seen that the graph representations for the transportation problem, maximal flow problem, and the optimal assignment problem are good candidates for modelling the IS problem as they have similar graph representations to the IS graph representation, outlined in Chapter 4, and will require slight modifications to facilitate the application of the algorithms identified, viz. the Hungarian method, 2-Phase Simplex method and the Edmonds Karp-algorithm.

5.1 The transportation problem and the 2-Phase Simplex method (SM)

5.1.1 A gentle introduction to linear programming

Linear programming (LP) is a mathematical discipline concerned with finding the optimal value of a linear function that is subject to a number of linear functional constraints (Thie and Keough, 2011). The formal representation of a standard linear program is shown below:

$$Maximize\ Z = \sum_{i=1}^{n} C_i X_i \qquad \text{Equation 5-1}$$

$subject\ to$: $\sum_{j=1}^{n} a_{1j}x_j \leq b_1$

$\sum_{j=1}^{n} a_{2j}x_j \leq b_2$

$\vdots$

$\sum_{j=1}^{n} a_{mj}x_j \leq b_m$

$x_1, x_2, \ldots\ldots, x_n \geq 0$

where there are n decision variables and m linear constraints, called functional constraints, and n non-negativity constraints $x_j > 0\ \forall\ j$. Linear problems with all functional constraints being $\leq$ and variables being restricted to non-negative values are said to be in the standard form (Hillier and Lieberman, 2010).

A simple standard linear program (SLP) may be used to understand the components of a linear program and how they relate to each other. Consider the following:

$$Maximize\ Z = x_1 + x_2$$

$$subject\ to:$$

$$2x_1 + x_2 \leq 16$$

$$x_2 \leq 4$$

$$x_i > 0, i = 1,2$$

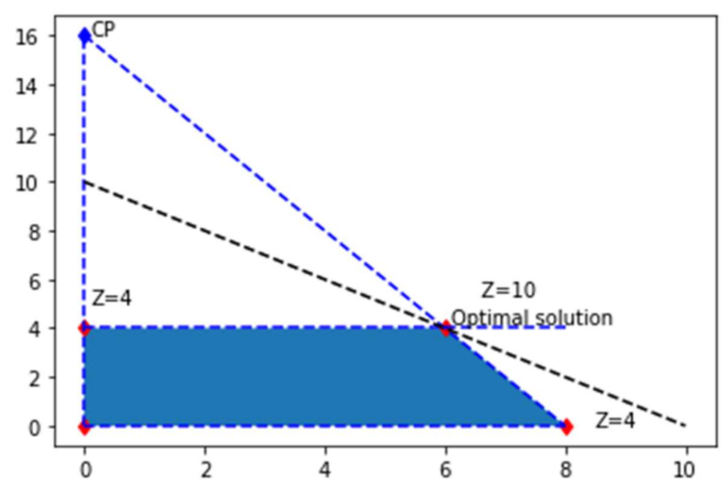


Figure 5-1: Geometric representation of simple linear program.

Figure 5-1 shows the geometric representation of the simple linear program that has two decision variables. The dotted blue line segments are the functional constraints. There are two decision variables (x_1 and x_2) and the blue-coloured region is what is known as the feasible region, which is bounded by the functional constraints. It is within the feasible region, and at the boundaries of the region, that the solutions to the linear program are found and feasible with respect to the functional constraints. However, they may be different objective function values. Clearly, the maximum value of Z, or minimum $-Z$ value, that still satisfies all the constraints, is at $(6,4)$, which renders the highest Z value of 10 units. The black dotted line is the objective function $Z = f(x_1, x_2) = 10$.

Corner-points (CP) are those points that correspond to the intersection of two or more functional constraints. Corner-point feasible solutions (CPF) are those CP solutions that fall on the boundary of the feasible region and are identified by the decision variables.

At a boundary line, the corresponding functional constraint is in the form of an equality. It may be stated that for constraints of the form $\sum x_i \leq b$, Equation 5-2 may be substituted:

$$\sum x_i + x_{sl_j} = b_j \qquad \textit{Equation 5-2}$$

The variable x_{sl_j} is the j^{th} slack variable for the j^{th} constraint and is defined to be the extent to which the left-hand side of the original inequality is less than the right-hand side.

The addition of slack variables to all functional constraints of the form $\leq$ augments the linear program by these supplementary variables (Hillier and Lieberman, 2010). This converts the linear program into its augmented form, with the addition of m slack variables to the m functional constraints. As such, CPF solutions of the form $(x_i, x_{i+1}, .., x_n)$ are now better represented as an augmented solution $(x_i, x_{i+1}, .., x_n, x_{n+1}, .., x_{n+m})$ where the numerical values of the m slack variables are included. This is known as a basic solution and is an augmented CP solution. A basic feasible solution (BF) is an augmented CPF solution, with the only difference between the two being the inclusion of the numerical values of the slack variables in the former.

With the addition of the slack variables, there are $m + n + 1$ variables and $m + 1$ equations; the objective function is included as the additional equation, with the corresponding variable being Z. This means there are n free variables that can be assigned arbitrary values.

This means that by assigning an arbitrary value to n variables, the values of the m other variables may be determined. These n variables are called non-basic variables and the m variables that are used to solve the m equations are the basic variables (known as the basis). The distribution of the slack and decision variables as basic or non-basic depends on BF solution, with adjacent BF solutions having all but one of their non-basic variables being the same. A BF solution will always have non-negative basic variables, regardless of whether the variables are slack or a decision variable. The n non-basic variables are set to zero, an arbitrary value commonly used in the SM (Hillier and Lieberman, 2010).

The optimal solution to the linear program will be at one of the extreme points/CPF solutions. In fact, for any linear program, the optimal solution will always be at the intersection of the line segments defined by the functional constraints. This is true due to the following properties of CPF/BF solutions:

1. If there exists exactly one optimal solution, then that solution corresponds to a CPF solution.
2. There are a finite number of CPF solutions.
3. A CPF solution is the optimal solution if there exists no adjacent CPF solutions that return a larger value for the objective function Z. (Hillier and Lieberman, 2010)

Once an SLP is converted into its augmented form, the Simplex method (SM) can be used to solve the linear program, to produce an optimal solution. It is a linear programming tool developed by George Dantzig in 1947 (Hillier and Lieberman, 2010). The SM solves the linear program by visiting a sequence of BF solutions and evaluates if the current BF solution is optimal or not. If not, it moves to the adjacent CPF/ BF solution that returns a higher objective function value.

The difficulty is to identify the initial BF solution from which the SM starts at. A common approach is to set the slack variables as the basic variables and the decision variables as the non-basic variables (Hillier and Lieberman, 2010). In this way, the initial BF solution is at the origin (since all decision variables have a value of zero), from which the SM will proceed to find the optimal BF solution. Hence, for an SLP, the initial m basic variables are the m slack variables.

5.1.2 The transportation problem

The transportation problem (TP) is a linear programming problem that is concerned with the distribution of a commodity from a set of suppliers to a set of consumers in such a way that the total cost of transportation is minimized (Hillier and Lieberman, 2010). This would be an example of an ECD policy. The graph representation of the transportation problem is shown in Figure 5-2, with x_{ij} being the edge between vertex i (Supplier) and vertex j (Consumer). The variable k_{ij} is the weight of the edge between supplier i and consumer j and represents the transportation cost per unit commodity exchanged between supplier i and consumer j.

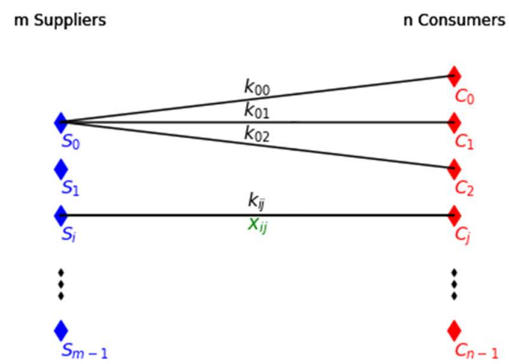


Figure 5-2:Graph-theoretic representation of the Transportation problem for an m×n system.

The total cost of transportation is given in Equation 5-3, the objective function of the linear program of the TP for an $m \times n$ system. The decision variable $x_{ij}$is the amount of resource exchanged between supplier i and a consumer j. The functional constraints are given in Equation 5-4, where σ_{S_i} is the resource amount supplier i has and $\sigma_{C_j}$is the resource amount consumer j needs.

Minimize:

$$Z = \sum_{i=0}^{m} \sum_{j=0}^{n} k_{ij} x_{ij} \qquad \text{Equation 5-3}$$

subject to:

$$\sum_{j=0}^{n} x_{ij} \le \sigma_{S_i} \ \forall\, i \,\epsilon\, [0, m] \qquad \text{Equation 5-4}$$

$$\sum_{i=0}^{m} x_{ij} \ge \sigma_{C_j} \ \forall\, j \,\epsilon\, [0, n]$$

$$x_{ij} \ge 0 \ \forall\, i, j$$

The constraints explain that the total amount of commodity sent from any supplier i to all consumers ($j \in [1, n]$) can't be greater than the supplier's availability. On the other hand, the total commodity supplied to a consumer j by all suppliers ($i \in [1, m]$) can't be less than that demanded by the consumer. With this in mind, what is sought is the minimisation of the objective function Z, whose value is proportional to the variables x_{ij} and the corresponding cost coefficient k_{ij}. Hence, the values of the variables x_{ij} are optimal when the objective function Z is minimised.

The TP is not an example of an SLP, since not all the functional constraints are of the form $\leq$. As such, the initial BF solution will be hard to identify. A standard approach is the artificial variable technique (Hillier and Lieberman, 2010), where an artificial version of the problem is created after modification of the objective function and transformation of the functional constraints of the form $\geq$:

$$x_1 + x_2 \geq b \rightarrow x_1 + x_2 - x_{su} + x_{av} = b$$

The addition of a surplus variable x_{su} signifies the extent to which the left-hand side is in surplus of the right-hand side and allows for the inequality constraint to be converted to an equality constraint. The artificial variable x_{av} is introduced as the "slack variable" for $\geq$ inequalities. Hence for t $\leq$ inequalities and v $\geq$ inequalities, there would be $t + v$ initial basic variables with a value of 0 (the initial BF solution) and n non-basic variables (the n decision variables). Surplus variables can't be an initial basic variable since it would violate the non-negativity constraint, which is why the artificial variable is required. There are two SM approaches to solving these linear programs, the Big M method and the 2-phase Simplex method, which is a more streamlined version of the former method.

The Two-phase Simplex method (SM) is a linear programming tool that splits the linear program into 2 phases. In the first phase, the solution to the artificial problem is found, in which the artificial variables are included in the initial basis for Phase 1. At the end of Phase 1, where the optimal BF solution is found, all artificial variables will obtain a value of zero, and this BF solution becomes the initial solution to Phase 2. If any artificial variable has a non-zero value at the end of Phase 1, then no solution exists for the original linear program (Hillier and Lieberman, 2010). In phase 2, the original objective function is restored, and the standard Simplex method can be used to find the optimal BF solution to the real problem.

In this way, the addition of slack, surplus, and artificial variables to the TP can be used to obtain an optimal solution via the 2-Phase Simplex method. This makes the TP a special application of the 2-Phase SM.

Conversion of the TP into the Standard TP (STP)

A feasible solution to the TP is one where the constraints are satisfied. It is clear from Equation 5-3 and Equation 5-4that it is possible for more than one feasible solution to exist such as one where a consumer j can receive more than their demand σ_{C_j} ($\sigma_{C_j} < \sum_{i=0}^{m} x_{ij}$) and a supplier i give less than they have ($\sum_{j=0}^{n} x_{ij} < \sigma_{S_i}$). This is of course not a desirable feasible solution in the case of IS, as there are additional storage costs associated with a consumer receiving excess by-product material than is required. Hence, to make the constraints binding, i.e., to ensure a consumer receives at most what they require and promote the situation where a supplier gives all their inventory, the set of feasible solutions becomes narrow. This would require changing the functional constraints to equality constraints, which turns out to be particularly beneficial for the run time complexity.

The more functional constraints there are, the more slack and surplus variables there will be. Meaning, geometrically, there will be a higher density of CPF solutions in the space. This results in a high computational expense in solving the linear program, since the SM reaches the optimal solution to the problem by visiting adjacent extreme points in succession. Since the pivoting rule from Dantzig's formulation of the SM makes the SM an exponential time algorithm (Klee and Minty, 1972, Hillier and Lieberman, 2010), it makes sense to reduce the number of variables in the problem where possible. By changing all the functional constraints in the problem formulation to equality constraints, the need for slack and surplus variables is avoided, and the addition of artificial variables is the only necessity. In this way, the transportation problem is transformed into its standard form.

The time complexity of the algorithm is more sensitive to the number of functional constraints than the number of variables (Hillier and Lieberman, 2010). It is proportional to the cube of the functional constraints, with the number of variables influencing the complexity to a minor degree (Hillier and Lieberman, 2010). Nonetheless, while the transformation of the constraints to equality constraints may not change the order of the time complexity, it may change the multiplying factor.

Due to the special structure of the mathematical formulation, a tableau representation is commonly used. The simplex table (ST) for an $m \times n$ system consists of $m \times n + 1$ rows (one for each constraint and the objective function equation) and $m \times n + m + n + 2$ columns (for the Z variable, $m \times n$ decision variables, $m + n$ artificial variables, and the right-hand side values).

For a 2×2 system, the simplex table for the transportation problem is compactly represented in an ST as :

	Z	x_{11}	x_{12}	x_{21}	x_{22}	x_{a1}	x_{a2}	x_{a3}	x_{a4}	**RHS**
Z	1	k_{11}	k_{12}	k_{21}	k_{22}	0	0	0	0	0
x_{a1}	0	1	1			1				θ_{S_1}
x_{a2}	0			1	1		1			θ_{S_2}
x_{a3}	0	1		1				1		$abs(\theta_{C_1})$
x_{a4}	0		1		1				1	$abs(\theta_{C_2})$

The variables $(x_{a1}, x_{a2}, x_{a3}, x_{a4})$ in the first column are the basic variables (the basis). The Simplex method makes use of the Simplex algorithm, which operates on the Simplex tableau and results in the change in basis until an optimal solution is reached. The basis is the minimal set of variables, called basic variables, that define the solution to the linear problem. Moving from one CPF solution results in a change of at least one basic variable to a non-basic variable and vice-versa. Phase 1 of the 2-phase Simplex method is the use of the Simplex algorithm in finding an initial feasible solution, whereas Phase 2 involves using the Simplex algorithm to find an optimal solution from the initial feasible solution. For a more detailed understanding of the 2-Phase Simplex method, the *Introduction to Operations Research* by Lieberman and Hillier(2010) is recommended. Figure 0-1 and Figure 0-2, shown in the appendix, represent the flow chart for the simplex algorithm and the simplex method for the transportation problem, respectively.

Balanced STP

For cases in which the total supply is not equal to the total demand, an additional "dummy" vertex is required to be added to the graph representation in Figure 5-2 and an additional equality constraint:

$$\sum_{j=0}^{n} x_{DummyNodej} = abs(Deficit)\ ,\ \ if \sum_{i=0}^{m} \sigma_{S_i} < \sum_{j=0}^{n} \sigma_{C_j}$$

$$\sum_{i=0}^{m} x_{i\ DummyNode} = abs(Deficit)\ ,\ \ if \sum_{i=0}^{m} \sigma_{S_i} > \sum_{j=0}^{n} \sigma_{C_j}$$

As a result, a balanced transportation problem is created, where $\sum_i \sigma_{S_i} = \sum_j \sigma_{C_j}$ (Total supply=Total demand)

To avoid departure from the original structure of the transportation problem, every edge incident to the "dummy" vertex is assigned a zero coefficient in the objective function, making it impossible for it being in the basis for the optimal solution.

$\sum_{j=0}^{n} k_{DummyNodej} x_{DummyNodej} = 0$ (if additional dummy supplier is added)

$\sum_{i=0}^{m} k_{iDummyNode} x_{iDummyNode} = 0$ (if additional dummy consumer is added)

Hence, for an $m \times n$ system, the transportation problem can be solved with the 2-Phase simplex method, after the system has been converted into a balanced TP, which is said to be its standard form.

5.1.3 Adaption of STP and SM to Industrial symbiosis

The coefficients in the objective function are used to reflect transportation costs in the TP. The larger $|k_{ij}|$ is, the stronger the supplier i and consumer j pairing is in the light of the simplex algorithm. The chief policy in the traditional STP is an ECD policy, which is a distribution based on the minimisation of transportation costs. For the application of IS, it is possible to reflect more than one policy by computing the objective function coefficients k_{ij} differently.

While the transport cost minimisation is inherent in the STP, the inclusion of an SCD policy is possible by factoring in the unit price (α) that each supplier and consumer attributes to the by-product material.

Essentially, suppliers and consumers who attribute similar economic value to the by-product material (similar worth considered) will lead to more stable and fair matchings since similar economic benefits to each company will be derived. A supplier and consumer who have a more similar value for the α product attribute (unit price) will reflect as a stronger pairing and, in this regard, will lead to the inclusion of an SCD policy. The policies are compactly represented using the $DistanceFactor_{ij}$ and $PriceFactor_{ij}$ which are, essentially, functions. The objective function coefficients k_{ij} are therefore adapted accordingly:

$$k_{ij} = -(PriceFactor_{ij} + DistanceFactor_{ij}) \qquad \text{Equation 5-5}$$

$$, PriceFactor_{ij} = MaxPrice - abs(\alpha_{S_i} - \alpha_{C_j})$$

$$DistanceFactor_{ij} = 1.1 \times MaxDistance - d_{ij}$$

$$MaxPrice = max\ ([\alpha]_k\ ), k\ \in S \cup C$$

$$MaxDistance = \max(\underline{D})$$

The matrix $\underline{D}$ contains the elements d_{ij} $(i \in S, j \in C)$, where d_{ij} is the distance of the shortest path between supplier i and consumer j. Section 5.4 elaborates on how this matrix is determined.

As was explained in Section 4.3, it is the subset of edges in the bipartite graph that represent the matching M and form the IS network. Since it is desired to find the optimal subset of edges from all possible supplier-consumer pairings, the value of all k_{ij}, $(i \in S, j \in C)$ must be nonzero. As a result, for all edge weights between the suppliers and consumers to be nonzero, an arbitrary coefficient of 1.1 is multiplied to the $Maxdistance$ value, ensuring $DistanceFactor_{ij}$ is never zero. Once this is done, the simplex algorithm will then favour the smallest coefficient in the objective function or a negative coefficient with the largest absolve value.

The Simplex method runs to optimality, ensuring that all the constraints are met and the value of the objective function is minimized. In other words, the resource requirements for all companies will be met in a way that results in the smallest objective function value.

5.2 Maximal Flow Problem and The Edmonds-Karp algorithm (FF)

5.2.1 An introduction to the Maximal flow problem and the Edmonds-Karp algorithm

The maximal flow problem is a network flow problem that was initially posed by T.Harris (Ford and Fulkerson, 2009) after his studies on the transportation problem. Historically, researchers became interested in the optimization of the flow of commodity after observing the Soviet railway system, with the idea of finding the maximum flow of goods possible from a source (starting location) to a sink (destination). The network that defines the paths from the source to the sink is the flow network. In the flow network G, the directed edge set E make up the paths and the vertex set V make up the locations. More especially, there are two distinct vertices, a single source vertex, and a single sink vertex.

The mathematical formulation is a linear program, with the optimality criterion being on the maximal flow of commodity through the network, rather than it being cost-based. The linear program is as follows:

Maximize

$$Z = \sum_{j=1}^{n} e_{0j} \qquad \text{Equation 5-6}$$

subject to:

$$\sum_{j} (f_{ij} - f_{ji}) = 0 \qquad i = 1, \dots . n-1 \qquad \text{Equation 5-7}$$

$$0 \leq f_{ij} \leq c_{ij} \qquad \text{Equation 5-8}$$

The parameter c_{ij} is the maximum flow (capacity) of commodity and f_{ij} is the current flow of commodity along the directed edge (arc) e_{ij}. A suitable analogy to explain the formulation is a pipe network. Equation 5-6 is a continuity constraint, in which the net flow entering a pipe junction (vertex) must equal that leaving the junction. Equation 5-7 is the capacity constraint, which can be used to model the constraints of an application. An example is when the flow in a pipe can't exceed the pipe capacity (c_{ij}) due to the volumetric size of the pipe or due to hydraulic considerations.

Figure 5-3 shows a flow network in which the source vertex is 0, and the sink vertex is 6. While the Simplex method can be adapted to solve this linear program, it is the Ford-Fulkerson method that describes the class of algorithms that solve the maximal flow problem (Lammich and Sefidgar, 2016). The Ford Fulkerson method was established by Dr. Fulkerson and L R Ford in 1956 to determine the maximal capacity of a network to carry flow from a source to a sink in a flow network (Ford and Fulkerson, 1956).

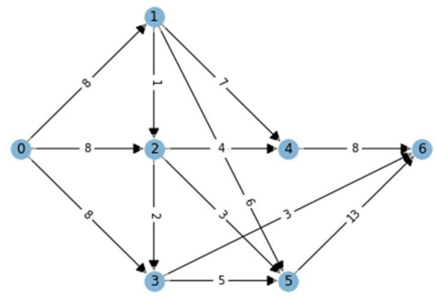


Figure 5-3: An example of a flow network, G.

The method operates on the residual network G_f of the flow network G, which has the same vertices as G and will contain arcs that have positive capacity values $c_{f_{ij}}$, called residual capacities. If a flow f_{ij} exists in G for arc e_{ij}, an arc e_{ij} is constructed in G_f with the value of $c_{f_{ij}}$ being $c_{ij} - f_{ij}$. An arc from j to i is constructed in G_f if an arc from i to j exists in G and has a residual capacity value of f_{ij}, the flow along e_{ij} in G. Symbolically:

$$c_{f_{ij}} = \begin{cases} c_{ij} - f_{ij} & if\ e_{ij} \in E \\ f_{ij} & if\ e_{ji} \in E \\ 0 & otherwise \end{cases} \qquad \text{Equation 5-9}$$

The residual graph G_f shows all possible pathways from the source vertex to the sink vertex with the maximum amount by which flow can be changed along each arc e_{ij} indicated by $c_{f_{ij}}$.

Figure 5-4: The flow network G (left) consisting of 1 arc with flow value f_{12}, and the corresponding residual graph G_f (right), with two arcs determined from arc e_{12} of G.

Figure 5-4 shows the residual graph of G(left), G_f (right). It can be seen that, while the maximum allowable flow for the arc e_{12} is a constant, (c_{12}), the flow value f_{12} becomes the adjustable parameter, as it is this parameter that is affected in the Ford-Fulkerson method.

It is noted that for simple flow systems, parallel arcs can't occur in the flow network G. Consequently, for simple flow systems, the residual graph G_f of G will have at most two arcs (e_{ij} and e_{ji} between any two vertices i and j) that are connected by an arc in G.

The next step is to find a path P in G_f from source to sink, called an augmenting path. The arc with the lowest available residual capacity is called the bottleneck of the path, as it is this arc that determines the maximum possible flow f_{ij} that can be added to each arc e_{ij} on the path in G without violating any of the constraints of the problem.

All flow values for the arcs in G are increased by c_p if they are in the augmenting path P, decreased by c_p if the reverse arc is in P or kept constant if the arc is not in P. Symbolically:

$$f_{ij_{new}} = \begin{cases} f_{ij} + c_p & if\ e_{ij} \in P\ and\ e_{ij} \in E \\ f_{ij} - c_p & if\ e_{ji} \in P\ and\ e_{ij} \in E \\ f_{ij} & otherwise \end{cases} \qquad \text{Equation 5-10}$$

, where:

$$c_p = \underbrace{min}_{e_{ij} \in P} (c_{f_{ij}}) \qquad \text{Equation 5-11}$$

The Ford-Fulkerson method can be summarized in the following way:

1. Create a residual graph G_f based on the original graph G using Equation 5-9
2. Choose an augmenting path from source to sink
3. Determine the minimum residual capacity c_p in the augmenting path using Equation 5-11
4. Update the flow values for all arcs in G using Equation 5-10
5. Recompute the residual graph G_f based on G using Equation 5-9
6. If there exists at least one augmenting path, return to step 2, otherwise the process is terminated.

Since there could be many augmenting paths from the source vertex to the sink vertex, the choice of the augmenting path becomes arbitrary. In choosing an arbitrary path, the flow of the network is increased but limited to the maximum allowable flow in the chosen path. Hence, a suboptimal solution could result because another path could have determined a larger increase in the flow of the network had it been chosen first. Therefore, it is the lack of structure in finding an augmenting path that makes the Ford-Fulkerson method a greedy approach and has been found to affect the computational efficiency of the Ford-Fulkerson method and the solution rendered (Cormen et al., 2009).

There are several heuristics for choosing augmenting paths, with each heuristic being better suited than the others for graph G based on the nature of the graph. The result of using a heuristic is that an algorithm is made available that essentially creates the order of the choice of augmenting path to consider. The Edmonds-Karp algorithm (FF) is a more efficient network flow algorithm than the Ford Fulkerson method as it uses the breadth-first search algorithm (BFS) for finding augmenting paths, and hence it is said to be an implementation of the Ford-Fulkerson method (Cormen et al., 2009).

While depth-first search (DFS) can also be used to determine the order of choosing augmenting paths with the same run-time as using BFS (Cormen et al., 2009), there are cases in which DFS results in an exponential runtime, and BFS does not. Consider the flow network in Figure 5-5 with source vertex 0 and sink vertex 3. The worse-case scenario is that the flow is increased by 1 unit. The DFS algorithm will take 2000 steps to produce the maximal flow. However, the BFS algorithm will take two steps. The Edmonds-Karp algorithm uses BFS to find the shortest augmenting path in G_f from source to sink, given the arc flow values of G. Hence, paths with the lowest number of edges from source to sink will be chosen as the augmenting path P first.

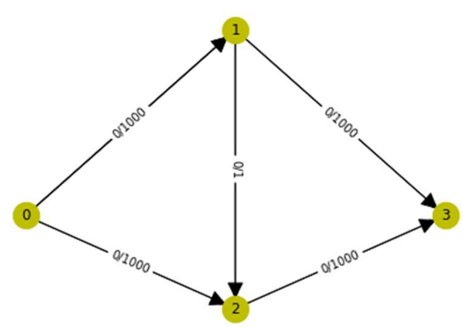


Figure 5-5: An example of a flow network for which the BFS is more suited than DFS.

5.2.2 Adaption of the Edmonds-Karp algorithm to IS

A key feature in the graph representation is that there is a single source vertex and a single sink vertex, with the objective being to push as much flow from source to sink. For the case of IS, where there are multiple sources and sinks, two additional vertices need to be added, one being a “super-source” and the other being a “super-sink” (Tardos and Kleinberg, 2005).

This transforms the problem into the traditional graph representation that the Edmonds-Karp algorithm was designed to solve. Hence the graph representation for the Edmonds-Karp algorithm is a graph $G = (S^* \cup C^*, E)$ where S^* is the set of suppliers $(s_0, s_1 \dots . s_m)$ appended with the "super-source node" and C^* being the set of consumers $(c_0, c_1 \dots . c_n)$ appended with the "super-sink node".

For the maximal flow problems, all edge capacities are computed as follows:

$$c_{ij} = \min\left(\sigma_{S_i}, abs(\sigma_{C_j})\right), i \in [1, m], j \in [1, n] \qquad \text{Equation 5-12}$$

$$c_{supersource\ i} = \sigma_{S_i}$$

$$c_{j\ supersink} = abs(\sigma_{C_j})$$

The graph representation of the maximal flow problem in IS is shown in Figure 5-6. With the computation of the directed edge capacities following Equation 5-12, the saturation of any edge $(f_{ij} = c_{ij}, i \in S, j \in C)$ occurs simultaneously with the saturation of either edge $e_{supersource\ i}$ or $e_{j\ supersink}$, depending on whether the supply or demand is greater in value. Symbolically:

$$f_{supersoucei} = \sigma_{S_i} = C_{SSource_i}\ \forall\ i\ \epsilon[1, m], if\ \sum\sigma_{S_i} < \sum\sigma_{C_j} \qquad \text{Equation 5-13}$$

$$f_{jsupersink} = abs\left(\sigma_{C_j}\right) = C_{C_jSSink}\ \forall\ j\ \epsilon\ [1, n], if\ \sum\sigma_{S_i} > \sum\sigma_{C_j}$$

In this way, the flow conservation constraint is upheld. The algorithm will produce augmenting paths and update flow values until there exists no path from the super-source to the super-sink. This will occur if either the set of edges incident to the super-source or that which is incident to the super-sink are saturated. In both cases, an IS network will be produced with at least one set of companies' resource requirements met. In this regard, the Edmonds-Karp method ensures that a feasible IS network always results when run. It is important to note that the saturation of the edges that allow for the flow conservation constraint to be upheld is due to the flow being equal to the by-product flow, with each directed edge capacity (constraint) known *a priori* (Equation 5-12). In other words, the edge capacities c_{ij} are a function of the σ product attribute (resource capacity) with $f: min(\sigma_{S_i}, \sigma_{C_j})$.

When one wants other policies to be considered, the implementation of the Edmonds-Karp algorithm becomes compromised. Consider the case where one seeks to prioritize pairings between companies who are geographically closer (ECD policy) and their unit price proposals (α) are most agreeable (SCD policy. The edge capacities will be computed as follows:

$$c_{ij} = Pricefactor_{ij} \times \min\left(\sigma_{S_i}, \sigma_{C_j}\right) + Distancefactor_{ij}, \qquad \text{Equation 5-14}$$

$$i \in [1, m], j \in [1, n],$$

$$PriceFactor_{ij} = MaxPrice - abs(\alpha_{S_i} - \alpha_{C_j})$$

$$DistanceFactor_{ij} = (1.1 \times MaxDistance - d_{ij}) \times RPD$$

If an augmenting path is chosen such that supplier k exchanges by-product material with consumer l, with f_{kl} increased from 0 to c_{kl}, it is either the supplier's resource amount that is diminished or the consumers' demand that is met, depending on which is smaller in absolute value. If $\sigma_{S_k} < abs(\sigma_{C_l})$, then the edge e_{kl} will be saturated. As a result, there can be no future pairings between this supplier k and any other consumer. However, the graph property that ensures this, namely the mutual saturation of edge $e_{supersource\ k}$ and e_{kl}, is, in general, not present since for mutual saturation of the edges to occur:

$$c_{supersource\ i} = Pricefactor_{kl} \times min(\sigma_k, \sigma_{C_l}) + Distancefactor_{kl}\ , k \in [1, m], l \in [1, n]$$

While the terms on the right are known, and σ_{S_k} and σ_l can be solved for and updated accordingly, all other $c_{supersource\ i}$, c_{ij} and $c_{j\ supersink}$ $(i \in [1, m], j \in [1, n])$ must be re-evaluated, using Equation 5-9. This is so that when the next augmenting path is found involving either supplier k or consumer l, the right-hand side of Equation 5-14 is equal to its left-hand side. Failing to update the edge capacities will result in the incorrect amount of commodity being sent from a supplier to a consumer, since f_{ij} is no longer a sole function of σ_{S_i} and σ_{C_j}.

Essentially, the graph G must be recomputed after each augmenting path is found, corresponding to the exchange of by-product material between a supplier and a consumer. As a result, a modification in the Edmonds-Karp method has to be made. Figure 0-3 shows the flowchart for the general implementation of the Edmonds-Karp algorithm for maximal flow problems, and Figure 0-4 shows the modified implementation of the algorithm in its application in IS for transportation cost minimization.

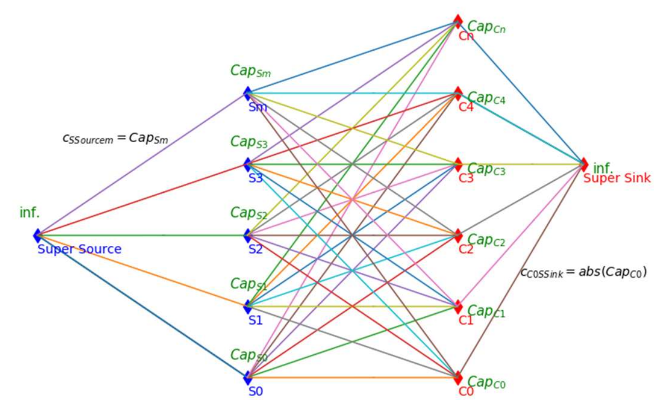


Figure 5-6: Graph representation of IS in maximal flow problems.

5.3 Optimal Assignment Problem and the Hungarian method (HM)

5.3.1 Introduction to the Optimal Assignment problem and the HM

The assignment problem is a Linear Programming problem that is concerned with assigning a set of n objects to another set of n objects in the way that is optimal. The assignment problem is a minimization problem that has been considered as a special case of the transportation problem (Hillier and Lieberman, 2010).

The standard form for the assignment problem is as follows:

$$Minimize\ Z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$subject\ to$: $\quad \sum_{i=1}^{n} x_{ij} = 1 \quad for\ i \in [1, n]$

$\sum_{j=1}^{n} x_{ij} = 1 \quad for\ j \in [1, n]$

$x_{ij} \geq 0\ \forall\ i, j$

$x_{ij} \in (0,1)\ \forall\ i, j$

Stated informally, the assignment problem can be viewed as there being n jobs available in a company and n employees looking to fill a position in the company. There is a cost that the company incurs in assigning an employee to any one of the jobs available. The problem poses the question about which job an employee should be assigned to such that the cost of the assignments is minimized. Since an employee can either be assigned to a job or not, the variables x_{ij} are restricted to binary values (0 or 1), making the assignment problem a binary integer program.

A matrix $\underline{B}$ can be used to represent this scenario, with each element in the matrix being b_{ij}. The element b_{11} is the cost of assigning individual 1 to job 1. The matrix $\underline{B}$ is also an assignment matrix, where one set of elements (suppliers/employees) are ordered along the rows, and the other set of elements (consumers/job positions) are ordered along the column.

$$\underline{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{bmatrix}$$

The graph representation for the assignment problem is a bipartite graph $G = (V, E), V = V_1 \cup V_2$. Following the analogy, one set of vertices represents the employees (V_1) and the other set (V_2) represents the jobs. The edge e_{ij} corresponds to the entry b_{ij} in the matrix $\underline{B}$ whose value represents the weight of the edge e_{ij}, which is the cost of matching a vertex from V_1 to a vertex from V_2. A bijection is the solution output, where exactly one employee is matched to a job.

Several researchers have worked on creating an algorithm for the assignment problem. The Hungarian method (HM) was formulated by H.W Kuhn, based on the work by Hungarian researchers D. König and J. Egerváry (Kuhn, 1956). Kuhn was studying the graph theory book by Dénes König when he encountered the author's augmenting path algorithm for the matching problem on bipartite graphs (Martello, 2010). Kuhn realized that the matching of a bipartite graph on two sets, each with a size of n, was identical to an n by n assignment problem with all decision variables $a_{ij} = 0 \; or \; 1$ (Schrijver, 2005). Based on a footnote in König's book, Kuhn was directed to a paper by J. Egerváry that contained the method in which a general assignment problem could be reduced to a finite sequence of 0-1 assignment problems (Kuhn, 2012).

Based on their work, the Hungarian method was formulated. A few definitions need to first be established before understanding how the HM works.

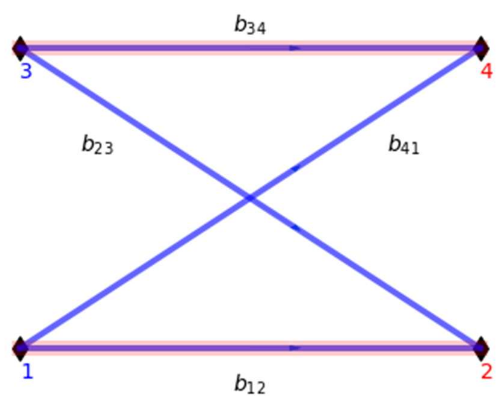


Figure 5-7: Bipartite graph showing a cycle C (blue edges) and a matching μ (highlighted edges).

Consider the bipartite graph in Figure 5-7. The blue edges (b_{12}, b_{23}, b_{34} and b_{41}) make up a cycle C, which is a path that starts and ends at the same vertex (v_1, v_2, v_3, v_4, v_1). The subset of edges highlighted in pink may be defined as a matching μ. The blue edges may be regarded as the potential matches, and the pink highlighted edges represent the current pairings. Cycle C is said to be μ-alternating if every other edge of C is in the matching μ (Roughgarden, 2016), which happens to be the case ($\boldsymbol{b_{12}}, b_{23}, \boldsymbol{b_{34}}$ and b_{41})

C is a negative μ-alternating cycle if edges in the matching have a higher total cost than those outside the matching:

$$\sum_{b_{ij} \in C \cap \mu} b_{ij} > \sum_{b_{ij} \in C \setminus \mu} b_{ij}$$

The matching μ in Figure 5-7 is a bijection defined by the edges $C \cap \mu$. Note that $C \setminus \mu \equiv \acute{\mu}$ is also a bijection, connecting the same vertices in a different way. Lastly, one can define a path from u to w in the graph G as a good path if:

- Both u and w are unmatched, $u \in V_1, w \in V_2$.
- The path contains edges in μ alternating with edges not in μ.
- Every edge of the path has a residual capacity $c_{f_{ij}}$ of zero, as defined in Equation 5-9, where $c_{ij} = b_{ij}$. These edges can be defined as "tight" edges (Roughgarden, 2016).

With these definitions out of the way, the HM algorithm can be understood using the following higher-level structure (Roughgarden, 2016):

<u>Hungarian method</u>

while there is no bijection ***do***

    if *there is a good path P* ***then***

        augment μ by P

    else *find a good set S and update the $\underline{B}$ matrix elements accordingly.*

The augmentation step is done using the ring sum operation $\mu \oplus P$, which creates a new graph with edges that are either exclusively in μ or P but not both. Finding a good path is done using the BFS algorithm, which is used to construct a tree. The BFS algorithm starts at an unmatched vertex v. Odd layers of the BFS tree consist of vertices that are connected with vertices in the previous layer via a tight edge but are not matched to these vertices. Even layers of the BFS tree are matched to vertices in the next layer of the BFS tree. Hence, the BFS tree will contain a good path P, if there is a μ-alternating cycle, else the elements in the matrix $\underline{B}$, of size $r \times r$, will be updated.

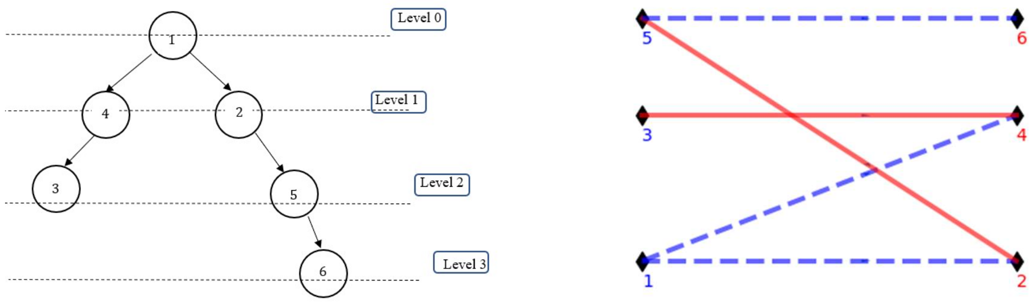


Figure 5-8: BFS tree (left) for the bipartite graph (right) with matchings (red graph edges) and tight edges (blue dashed lines).

Figure 5-8 demonstrates this procedure, with all edges in the graph being tight and matched vertices being connected by red edges. The good set S is made up of the vertices in the even layers of the BFS tree (v_1, v_3, v_5). NS can be defined as the set of vertices at the odd levels of the BFS tree (v_4, v_2, v_6). Hence, the price updation step is:

for *all u ∈ S* ***do***

decrease b_{uk} *by* Δ

for *all w ∈ NS* ***do***

increase b_{kw} *by* Δ

, where $k \in [1, r]$, $\Delta = \min(b_{ij})$, $b_{ij} \in \underline{B}, b_{ij} > 0$.

In this way, new edges become tight until the algorithm terminates when a bijection is an output as a result of the ring sum operation. The flowchart of the general implementation of the Hungarian algorithm is shown in Figure 0-5.

5.3.2 Adaption of the HM to IS

The problem can be viewed in the light of IS, where an optimal matching is desired, with each matching having a certain cost. The smaller the cost of the matching, the more desirable it is. The minimization of by-product material that is sent to the landfill, for instance, can be achieved by computing the cost matrix $\underline{B}$ in the following way:

$$b_{ij} = abs(\sigma_{S_i} + \sigma_{C_j}) \qquad \text{Equation 5-15}$$

Hence, the smaller b_{ij} is, the more desirable the matching is and results in the minimization of the amount of virgin material bought by the supplier.

To include other policies, like the previously described SCD policy and ECD policy, the edge capacities can be computed as follows:

$$b_{ij} = abs\left(\alpha_{S_i} - \alpha_{C_j}\right) \times abs\left(\sigma_{S_i} + \sigma_{C_j}\right) + d_{ij} \times RPD \qquad \text{Equation 5-16}$$

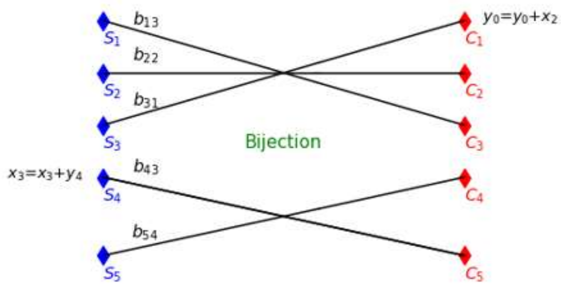
An example of the solution output from the Hungarian method is shown in Figure 5-9, corresponding to the highlighted elements in matrix $\underline{B}$. For each pairing, the resource capacities for the supplier is reduced and increased for the consumer, resulting in a net decrease in the total demand and total supply. Since a square matrix is the required input to the HM, if $n \neq m$, then a $p \times p$ assignment matrix $\underline{P}$ can be created, where $p = \max(n, m)$ with each p_{ij} in the assignment matrix $\underline{P}$ defined as follows.

$$p_{ij} = \begin{cases} b_{ij}, & if\ i \leq m, j \leq n \\ K, & otherwise \end{cases} \qquad \text{Equation 5-17}$$

The value K is an arbitrary large number, where $K \gg \max\left(b_{ij}\right), b_{ij} \in \underline{B}$, which reflects a weak pairing. Conceptually, this corresponds to adding $|m - n|$ dummy suppliers (if $m > n$) or $|m - n|$ dummy consumer (if $m < n$) with the matrix entry of K. The pairing made with a dummy supplier or consumer will be a weak pairing and will not factor in M, the matching in the IS network.

Unlike the Simplex method and the Edmonds-Karp method, one successful run of the HM does not guarantee an optimal IS network since the case may exist where some suppliers still have resource available and some consumers still being in need. Hence, the HM must be called iteratively until either the total demand or supply is met, whichever is the smallest in absolute value. In between each iteration, the bijection from the previous iteration must be stored and reflected in the matching M, as it will affect the pairings in the next iteration.

Consequentially, the subset of suppliers $\acute{S}(\acute{S} \subset S)$ whose supply is diminished will be taken out from the set of suppliers S when creating the new assignment matric $\underline{B}_{new}$ such that the bijection output from this assignment matrix will not pair any consumer with those suppliers. The same goes for those consumers whose resource demands are met. The step to creating matrix $\underline{P}$, shown in Equation 5-17 can then be proceeded with accordingly. The flowchart of the HM application in IS is shown in Figure 0-6.



$$\underline{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} \end{bmatrix}$$

Figure 5-9: Solution output from the Hungarian method corresponding to the optimal entries in matrix $\underline{B}$.

5.4 Dijkstra's algorithm and the Minimum Spanning tree

The Dijkstra's algorithm is a useful method for the determination of the relative distances between each supplier and each consumer. The algorithm will find the shortest path from a single source vertex to any other vertex if a path exists (Dijkstra, 1959). Dijkstra's algorithm solves shortest path problems consisting of a single source on a weighted graph $G = G(V, E)$ with non-negative weights (Cormen et al., 2009). Dijkstra's algorithm relies on the optimal substructure property, which is exhibited in problems whose optimal solution is made up of optimal solutions of subproblems (Cormen et al., 2009). The shortest part problem has the optimal substructure property in which the shortest part between two vertices is made up of other shortest paths within it. It is a greedy method as it always adds the next "closest" vertex to S that is closest to the source vertex.

The algorithm maintains a set of vertices S, with the final shortest paths from each vertex in S to the source vertex already determined. A vertex $u \in V - S$, with the minimum shortest path distance from the source, is repeatedly added to S until $V - S$ is empty. In this way, the shortest path from the source vertex to any other vertex in the graph is deduced by Dijkstra's algorithm.Consider a simple 3×3 system. The geographical locations of the three suppliers and consumers are shown in Figure 5-10. It is desired for the matrix $\underline{D}$ to be computed. So that d_{ij} is not just the straight-line distance from supplier i to consumer j, a road network can be generated to reflect a more realistic geographical setting. A convenient structure for the road network is by constructing the minimum spanning tree (MST) of the graph.

The MST is a tree of graph G that connects all vertices of G (Wilson, 2015). The MST of G is a spanning tree whose total edge weight is a minimum. Prim's algorithm was used to determine the MST. Prim's algorithm works in a similar manner to Dijkstra's algorithm for finding shortest paths in a graph (Cormen et al., 2009) as it maintains a set A of edges that form a single tree. The algorithm iteratively adds the lightest edge that connects an isolated vertex to the tree until the three spans all the vertices in

G. It is also a greedy method as, at each iteration, it includes edges that add the minimum weight to the edge weight of the tree (Cormen et al., 2009).

Based on the MST, which is a convenient structure for a road network, Dijkstra's algorithm can be used to determine the shortest distance between any supplier to any consumer by assigning the source vertex as a supplier. In this way, a path between any two vertices is guaranteed, and the distance matrix $\underline{D}$ can be computed using Dijkstra's algorithm where for element $d_{ij}, (i \in S, j \in C)$, the source vertex as input to Dijkstra's algorithm would be the vertex for S_i, which would deduce $d_{ij}\ \forall\ j \in C$. Flow charts for the Prim's algorithm and Dijkstra's algorithm can be found in the Appendix as Figure 0-7 and Figure 0-8, respectively.

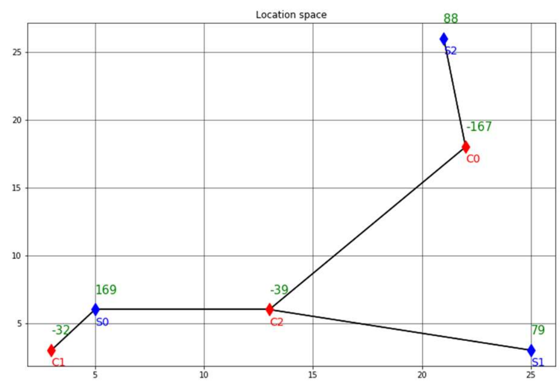


Figure 5-10:MST for a 3 by 3 IS system.

6 IS NETWORK GENERATION USING THE ALGORITHMS

Now that the graph representations for each network flow problem have been established, with the adaption of each network flow algorithm to the application of Industrial Symbiosis, it is possible to demonstrate the suitability of the algorithms in generating a feasible IS network. A single scenario is created where there are six suppliers and six consumers of a single resource that desire to be part of an industrial symbiosis. Each company is looking to have their resource requirement met such that the least amount of by-product material is sent to the landfill while keeping transportation costs as low as possible. In this way, an ECD and EVD distribution policy is favoured.

Companies that are paired may agree to share the transportation costs in the interest of fairness. To speed up the negotiation process, all companies require that the selling and buying unit price α for the by-product be as similar as possible between the supplier and consumer pairing. In this way, an SCD policy is included. A supplier may assert a unit resource price based on what allows them to make the IS venture a feasible solution and could be used to factor in as many of the operational, storage and administrative costs as possible. This could also reflect the product purity of the material and its perceived worth in its re-valorisation. For the consumer, the buying unit price that is volunteered would be one that fits the budget of the consumer after they have made an estimation of the expected purification costs (if any), transportation and storage costs.

The by-product material is chemically similar in nature amongst the suppliers. However, the approach can be implemented even when the resource is dissimilar in chemical nature but affords the functionality in its substitution for the consumers' raw materials.

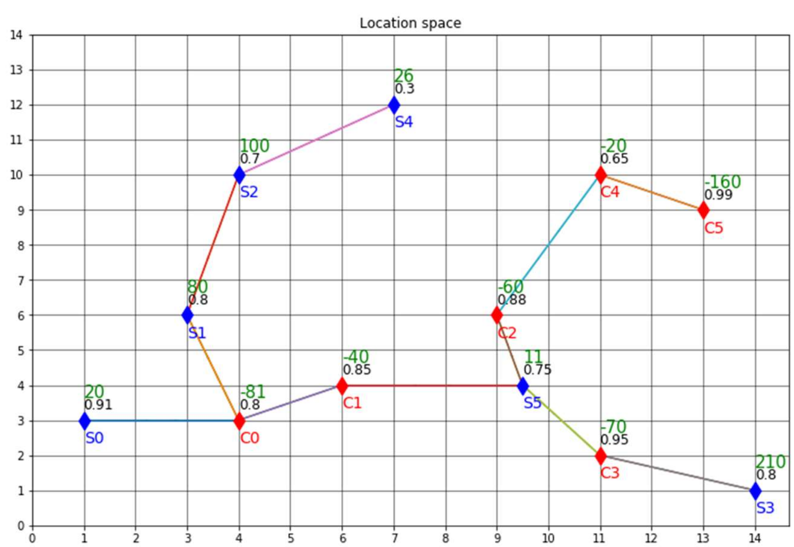


Figure 6-1: The minimum spanning tree for the 6× 6 system.

The product attributes for the set of suppliers and consumers (σ, α, θ) are provided in Table 6-1 and Table 6-2, respectively. The geographic location is conveniently represented as cartesian coordinates.

Table 6-1: Product attributes for the set of suppliers.

Product attribute	**Supplier**					
	0	1	2	3	4	5
Res. Capacity (unit)	20	80	100	210	26	11
Unit Price (R/unit)	3	4.5	1.5	4.5	1	5
X position	1	3	4	14	7	9.5
Y position	3	6	10	1	12	4

Table 6-2: Product attributes for the set of consumers.

Product attribute	**Consumer**					
	0	1	2	3	4	5
Res. Capacity (unit)	-81	-41	-60	-70	-20	-160
Unit Price (R/unit)	4	3	2	1	2	4
X position	4	6	9	11	11	13
Y position	3	4	6	10	2	9

The minimum spanning tree for this 6×6 system, shown in Figure 4-1, is shown in Figure 6-1. It is along the MST edges shown in Figure 6-1 that the Dijkstra's algorithm will find a path from any supplier to any consumer, until the distance matrix $\underline{D}$ is computed. The total supply of available by-product material is 447 units and the total demand is 431 units.

With this information, the bipartite graphs for the FF and HM, guided by their respective graph representations, was computed as well as the creation of the simplex tableau for the SM.

Each algorithm was run and produced an output as an IS network. Figure 6-2 shows the IS network rendered by the HM, Figure 6-3 shows that the IS network rendered by the FF algorithm and Figure 6-4 shows the IS network rendered by the SM.

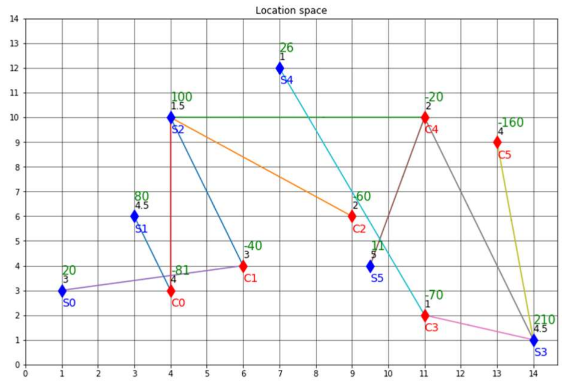


Figure 6-2: Industrial symbiosis network rendered by the Hungarian method.

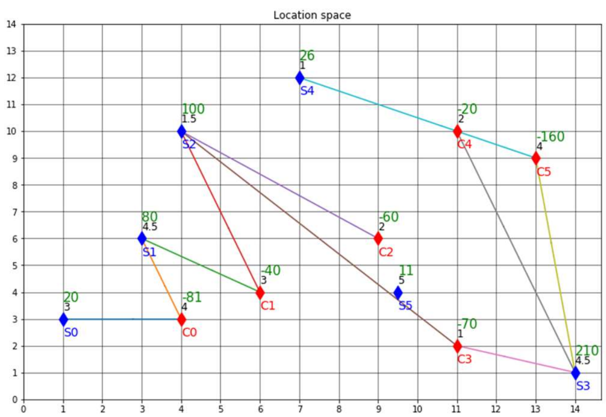


Figure 6-3:Industrial symbiosis network rendered by the Edmonds-Karp method.

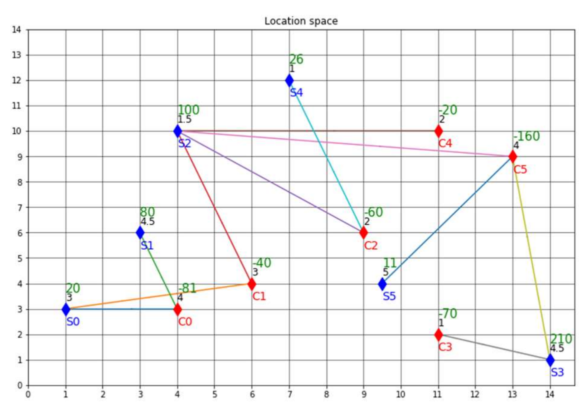


Figure 6-4: Industrial symbiosis network rendered by the Simplex method.

In all of the IS networks generated, the total demand is satisfied, confirming the fact that at least one set of companies' resource requirements will be met. The total supply is still greater than zero in all the IS networks generated since the total demand was less than the total supply. Multiple solutions exist for this problem, as is evident in the different IS networks produced by each algorithm. While the assignment matrix is the same in terms of the row and column order, with the product attributes for supplier i and consumer j being used to create element $(A_{ij})^{\prime}$ for each algorithm, there are differences in the network structure. This implies the following statement:

Statement 1: For a given assignment matrix $\underline{\acute{A}}: \acute{A}_{ij} = f(PA_{Si}, PA_{Cj})$*, the rendered IS network is dependent on the algorithm choice.*

If the assignment matrix A is computed as $\underline{\acute{A}}: \acute{A}_{ij} = f(PA_{Si}, PA_{Cj}), j \in [0,5], i = abs(5 - j)$, essentially transposing the assignment matrix, the IS network rendered by the Edmonds-Karp algorithm has a completely different structure than the one it previously produced (Figure 6-2), as seen in Figure 6-5.

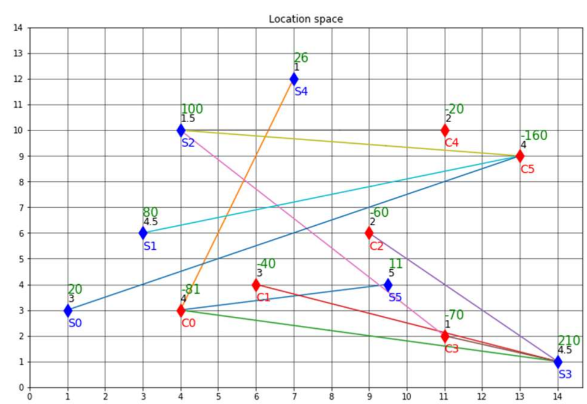


Figure 6-5: IS network rendered by the Edmonds-Karp algorithm with a different order of the rows and columns in the assignment matrix.

This implies the following statement:

Statement 2: For a given algorithm, the order of each company's product attributes in its respective product attributes list affects the IS network that is rendered.

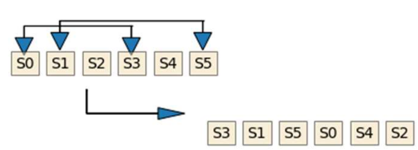


Figure 6-6: Illustration of the change in order in a product attributes list.

Statement 2 essentially talks to the order of the rows and columns for a given assignment matrix $\underline{\acute{A}}$. A different row and column order seem to produce different IS networks, revealing that multiple solutions exist for the IS problem. Furthermore, since the algorithms follow a specific routine when processing the assignment matrix, it is the order of the rows and columns that determines the order that supplier-consumer pairings are made.

Clearly, the larger the system, the larger the set of potential solutions, each with possibly different structural features. A way to concisely evaluate this sensitivity in the structural features of the IS network is to use graph metrics. Graph metrics can be used to differentiate between multiple optimal solutions and can be useful for conducting a comparative study of the algorithms and determining the optimality of solutions with respect to a distribution policy.

Graph metrics

A few basic graph measures are:

1. EdgeCount : number of edges in the graph
2. Vertex degree: Number of edges incident to a vertex
3. Number of graph components: The number of subgraphs
4. Total edge distance: Sum of all the edge weights, representing the sum of the pairwise travelling distances.
5. Eigenvector degree centrality: Vertex importance/influence (values determined for each vertex)

These metrics have significant implications to the IS matching characteristics and indicate anything from the "fairness" of allocations/opportunities for competitiveness to the environmental impact and profitability of the transportation companies involved in the network.

For example, in Figure 6-7 there are three graph components, which collectively make up graph G, and are demarcated by dotted boundaries. All the blue vertices have a vertex degree of 1 and the red vertices have a vertex degree of 2, except for the right most red vertex, which has three edges that are incident to it. There is a total of 7 edges, and if each edge has a unit length, the total edge distance is seven units.

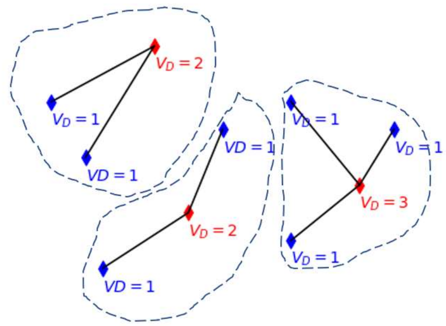


Figure 6-7: Illustration of subgraphs in a graph G.

One can imagine the red vertices being suppliers and blue vertices being consumers, in which case the metrics may be interpreted in the context of IS as follows: Three clusters are created in the IS matching, denoted by the 3 graph components, with smaller edge distances being better for companies in the cluster with respect to transportation costs. Higher edge counts may indicate there is generally more distribution across companies but also that transport activity and the associated environmental impact are greater.

However, the metrics could also indicate that opportunistic behaviour, such as price gouging, could result due to a reduction in transportation costs. The Eigenvector centrality for the red vertices would be calculated to be higher than that of the blue vertices, indicating that their level of influence in their respective clusters is greater. This has implications to the power dynamics of the IS network and has implications on network stability.

Hence, even though in Figure 4-1 a 6×6 system is shown and the application of any one of the algorithms will render a feasible IS network, one can ask the question, for or an $m \times n$ system, where $m, n \in \boldsymbol{N}$, which algorithm is most suitable for generating IS networks that are feasible with respect to the imposed distribution policy? With the insight into the bias that the order of the elements in product attributes lists and having defined the modelling of IS networks and the development of the graph representations for each algorithm, a foundation is laid for Chapter 7, which is concerned with the development of a methodology to assess the suitability of each algorithm in the creation of IS networks and to answer the research questions posed, as defined in Chapter 3.

7 METHODOLOGY BASED ON GRAPH METRIC ANALYSIS

In Chapter 6 it was seen that the product attributes were used to impose the distribution policies on the IS matching M and that the order of the rows and columns in the assignment matrix affects the order that the supplier-consumer pairings are made. This results in different IS network solutions, given that multiple solutions to the problem exist. From these 2 observations, an updated formalism is realised that generalises the matching problem for the IS application.

Given a cost ${r_{ij}}^{l}{}_{T}$, where ${r_{ij}}^{l}{}_{T}$ for the l^{th} product attribute in the pairing between supplier i and consumer j after T pairings have already been made, determine M, the set of pairings, such that Z is minimized, where:

Equation 7-1

$$Z = \sum_{l=1}^{|DV|} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} r_{ij_T}^{l}\, e_{ij}\, , T \in [0, \sup T], \qquad e_{ij} \in \mathbb{Z}_2$$

$$r_{ij_T}^{l} = f(PA_{Si}, PA_{Cj})$$

The product attributes DV are used to represent the policies that are the motivation for the industrial symbiosis. The binary variable e_{ij} is one if the pairing between supplier i and consumer j is in the matching or zero, otherwise. The formalism highlights the dependency that a pairing has on the order in which previous pairings are made. Since there are many different matchings that could be made, which may be suboptimal with respect to the policies, it is the correct order of pairings that results in the smallest Z value since $r_{ij_0}^{l}$ will have a different cost compared to $r_{ij_1}^{l}$; the previous pairing could have impacted the value of the l^{th} product attribute for supplier i and consumer j (such as the resource capacity σ).

With this formalism, an over-arching methodology was implemented that would allow answers to the research questions to be determined. This chapter lays out the framework and methodology used to carry out an investigation into whether any interesting insights can be gained from larger IS systems with the view of answering the research questions. The framework was created in the programming software Python and is made up of classes designed to be run in an integrated manner. The following are the main classes in the framework:

1. Dataset generation class
2. Graph Metric class
3. Algorithm class

Aside from these are functions/methods needed for pre-processing inputs to classes and post-processing of outputs from classes. The framework starts with the Data generation class, which creates d number of randomly generated datasets with m number of suppliers and n number of consumes. Each user (supplier/consumer) has the following product attributes ($l = 4$):

1. $\sigma =$ Resource capacity required to be shared
2. $\alpha =$ Unit price for the resource
3. $\varphi =$ Purity of the resource
4. $\theta =$ Geographic location of the resource

A supplier's resource capacity will be a positive value, and a consumer's resource capacity will be a negative value. The values of the product attributes for each company are randomised, within limits, and are shown in Table 7-2. As explained previously, the product attributes allow for distribution policies to be imposed on the Symbiosis, for example, the minimization of by-product material sent to landfill as waste, the minimization of transportation costs, or optimal by-product distribution among companies. Functions/methods are used for pre-processing the data for each user.

Based on the policy, a function $f(u, v)$ is applied for each pair u, v, where $u \in Suppliers, v \in Consumers$ resulting in a complete bipartite graph in the parameter space. This translates into all possible matchings between the set of suppliers and the set of consumers.

The algorithm, which is in the form of a class with its own methods, should output a graph in the form of an adjacency matrix, showing the matchings created. The graph representations for all the algorithms established in Chapter 5 will be used in this investigation. Hence, the same distribution policies imposed in the cost functions for each algorithm are the same one to be used in this investigation.

For the purposes of analysis, the graph metrics must be extracted and stored. This is done using a Graph metric class that extracts and records the relevant metrics required for analysis.

It was clear in the IS application shown in Chapter 6 that, for the case where multiple feasible solutions exist, the order the companies' product attributes (e.g., σ_{S_i}) are in their respective product attributes lists (e.g., σ_S) affects the IS network that is rendered. To investigate the apparent bias the order of the list elements and the choice of algorithm has on the rendered solution, a subroutine was created to randomize the order of the company's product attributes in their respective lists, as illustrated in Figure 6-6. This "shuffling" subroutine also records the new order of each company's product attributes in their respective lists, thereby allowing the order to be restored at a later stage, which is important to ensure consistency in the graph metrics. Performing this subroutine a number of times, for each dataset from the d data sets, necessitates that the graph metrics be reported as its respective mean values.

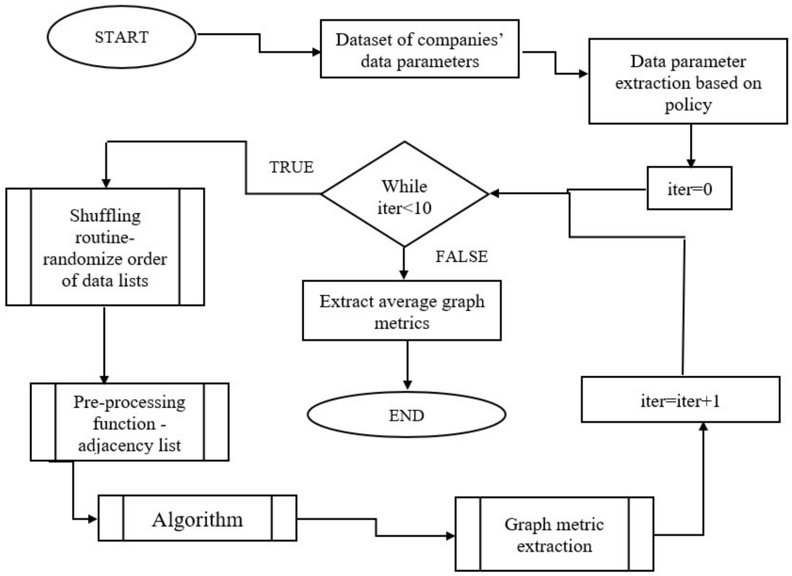


Figure 7-1:Flow diagram of the generalised framework implemented in the study.

Figure 7-1 illustrates the flow diagram of the framework. Ten was a convenient number of iterations to perform the shuffling routine to investigate the bias that the order of the elements in the product attribute lists has on the rendered solutions. Essentially, each algorithm is applied to a generated dataset and is performed ten times, with each time changing the order of the row elements and column elements of the assignment matrix $\underline{\acute{A}}$. Extending this to all d generated datasets is the entirety of this investigation.

These datasets can have more suppliers than consumers, fewer suppliers than consumers, or an equal number. With this in view, a scientific method can be used to structure the framework and investigation. The scientific method is concerned with how the datasets are generated. Following three blueprints, the generation of d datasets can be done in 3 ways:

Table 7-1: Table showing the 3 blueprints used for creating different sets of datasets, with each set having d datasets.

Dataset			Description
Name	Num. Supp.	Num. Cons.	
<u>GROUP SC</u>	C+ds_i, $i \in [1, d]$	C+ds_i, $i \in [1, d]$	Generation of d data sets, each with size $m \times n$, with $m = n$. The i^{th} dataset ds_i would have C+ds_i suppliers and consumers. Datasets generated in this way means that the number of suppliers and consumers are equal and are referenced as GROUP SC datasets, with SC indicating both suppliers and consumers are varied in number
<u>GROUP S</u>	C+ds_i, $i \in [1, d]$	τ	Generation of 50 data sets, each with size $m \times n$. The i^{th} dataset ds_i would have C+ds_i suppliers and τ consumers. The S in GROUP S indicates that suppliers are varied in number in the group of datasets
<u>GROUP C</u>	τ	C+ds_i, $i \in [1, d]$	Generation of 50 data sets, each with size $m \times n$. The i^{th} dataset ds_i would have C+ds_i consumers and τ suppliers. Every dataset had a constant number of suppliers. The C in GROUP C indicates that consumers are varied in number in the group of datasets

For each dataset, the product attributes for each company are given random values. The bounds on the randomized product attribute values for each company are given in Table 7-2. For ease of terminology, consumer-dominant datasets are those in which the number of consumers is greater than that of suppliers, whereas supplier-dominant datasets are those datasets where the opposite is true. The three blueprints for the generation of the datasets differ with respect to the number of suppliers to consumers. Each blueprint is used to generate a set of datasets, with the ***GROUP S datasets*** having the same number of consumers ($\boldsymbol{\tau}$) but differ in the number of suppliers.

The ***GROUP C datasets*** are those having the same number of suppliers ($\boldsymbol{\tau}$) but differ in the number of consumers. On the other hand, ***GROUP SC datasets*** are datasets in which the number of suppliers equals the number of consumers. In this way, different scenarios are created, distinct from each other by their size (number of suppliers and consumers) and which type of company is dominant in number.

Table 7-2: Bounds placed on the values of the product attributes.

Product attribute	Lower bound	Upper bound
Resource Capacity (units)	25	210
Unit Price ($R/unit$)	1	6
Geographical X-coordinate	0	25
Geographical Y-coordinate	0	30
By-product purity	0.455	0.999

Figure 7-2 shows the schematic of how Figure 7-1 fits into the study for the purposes of obtaining results. Figure 7-1 can be thought of as the main generalized routine, and Figure 7-2 shows how IS networks from each dataset type will be processed and analysed.

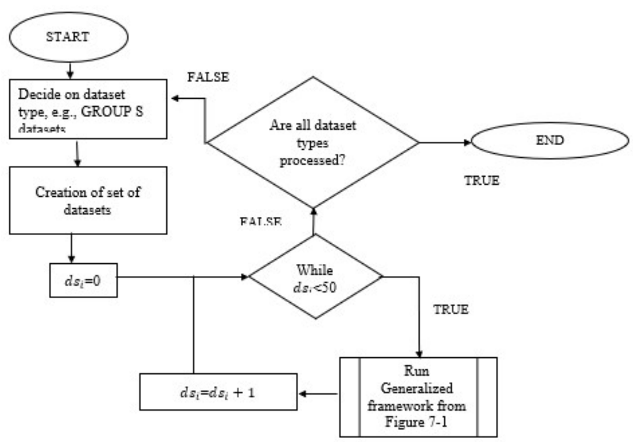


Figure 7-2:Flowchart of the overall methodology used in this study.

This methodology was carried forward, with IS networks being produced from the simulated data sets. The graph metrics results were obtained and are presented in Chapter 8, where results are visualised and summarised in the form of scatter plots and tables. Observable trends in the results are interpreted with respect to the distribution policies and in light of the research questions posed in Chapter 3.

8 SIMULATION RESULTS

The purpose of Chapter 8 is to analyse and interpret the results of the rigorous study, explained in Chapter 7. It was shown in Chapter 6 that all the three algorithms, i.e., the SM, HM, and FF create feasible IS networks with respect to different distribution policies. Furthermore, it was found that the IS network solution is dependent on the algorithm used, and the order of the product attributes in the lists used to create the assignment matrix, as illustrated in Figure 6-6. This observation was summarised in two statements (*Statement 1 and 2*) in Chapter 6. To explore the implications of these observations with larger datasets and thereby infer the suitability of the algorithms in creating IS networks, a rigorous study was devised using a methodology that creates and processes the results of sets of machine-generated datasets.

Each dataset containing the product attributes for a pre-determined number of supplier and consumer companies. The companies for a given dataset are keen to engage in an Industrial symbiosis of a single by-product material, with different distribution policies imposed as constraints on the distribution that is to occur. The size of the system for a dataset refers to the number of companies that are keen to engage in the IS. Once all company's product attributes are extracted from a machine-generated dataset and represented by the graph representation of algorithm i, the optimality of the rendered IS network (as a graph) can be determined with respect to the distribution policies. The optimality is interpreted by extracting and analysing the graph metrics for the rendered IS network. A set of datasets is defined as a list of datasets that differ in size, with succeeding datasets in the list being larger in size than the preceding dataset.

Processing a set of datasets by an algorithm i using the methodology in Figure 7-2 allows for the investigation of how the optimality of an IS network is affected by the size of the IS network. Furthermore, by processing the set of datasets using the different algorithms, i.e., the SM, HM and FF, independently, one can determine if there exists any superiority in IS network creation between the algorithms with respect to the distribution policies.

Three sets of datasets were created, as defined in Table 7-1, with the difference being in how the size of the system for consecutive datasets is increased. The size of the set is determined by the constant d (number of datasets). The constant τ is the number of supplier or consumer companies in a group and is constant across all iterations. The constant C defines the initial number of supplier or consumer companies at the first iteration in a group (e.g. GROUP S). Therefore, the values of τ and C define the size of the first dataset. **GROUP SC** datasets increase in system size by both the number of suppliers and consumers, with an equal number of the two types of companies in each dataset of this group. **GROUP S** datasets are those datasets with a different number of suppliers but the same number of consumers. In this way, one can determine how increasing the number of suppliers for a given number of consumers affects the IS network feasibility. **GROUP C** datasets have a different number of consumers but the same number of suppliers, and once they are processed, they can indicate how an increase in the number of consumers for a given number of suppliers can affect the IS network feasibility.

For this study, τ was chosen as 30, d as 50, and C as 5. For each set of datasets, 50 were created, differing by either the number of suppliers, consumers, or both, as shown in Table 7-1. These values allow for there to be datasets in which there is a sufficiently large discrepancy between the number of suppliers and consumers for **GROUP S** and **GROUP C** datasets. It is anticipated that in practical application, suppliers and consumers can be identified from different industries, especially when the geographical area of implementation is not confined to an EIP. Therefore, these values could be representative of practical application. The definitions of the product attributes are reproduced below:

1. $\sigma =$ Resource capacity required to be shared
2. $\alpha =$ Unit price for the resource
3. $\varphi =$ Purity of the resource
4. $\theta =$ Geographic location of the resource

Random values were generated for each product attribute for all the companies in each dataset (Table 7-2). These product attributes from all companies were then used in the graph representations for each algorithm, creating a bipartite graph.

The bipartite graph, represented as an assignment matrix $\underline{\acute{A}}$, was then processed by the respective algorithms, 10 times, with each time changing the order of the rows and columns of the matrix in a random manner. A value of 10 (i.e. 10 reshuffles) was deemed a convenient and applicable value to demonstrate the different number of feasible solutions that are possible for this case study while also being small enough so as to not result in substantially long computational run time, as highlighted in Figure 8-7. The output from the algorithm, together with its representation as an adjacency matrix, was recorded, from which the graph metrics were extracted and averaged over the ten runs. Hence, the output from a given dataset for a specific graph metric is three averaged graph metric values, one for each matching algorithm (HM, FF, SM).

At the outset, it was found that the key graph metrics in this study are the number of graph components and the number of graph edges. It is with these two metrics that the performance of each algorithm with respect to the distribution policies could be deduced and is, therefore, most stressed upon in the results. It must be reiterated that a subgraph is defined here as a graph made up of at least one edge and two vertices. Hence, an isolated vertex, having no incident edges, will not be classified as a subgraph.

For the results and analysis, it was intended to compare the statistical results of the graph metrics from each algorithm. However, to begin the analysis, the trends of the graph metric results will be interpreted by analysing the scatter plots of the results. Once an idea of the trends is gained, the statistical analysis will be used to verify the trends and how it relates to the quality of the connections and the IS Network, as a whole.

8.1 General trends observation

Since for each of the three sets of datasets (**GROUP S**, **GROUP C,** and **GROUP SC**) the size of the system changes across the datasets, it was decided to plot the graph metric results for the **GROUP S** and **GROUP C** datasets against the difference between the number of suppliers and consumers ($|m - n|$). In this way, a domain is created that allows for analysis into the effect of the difference in the number of suppliers and consumers has on the graph metrics of the IS network. To conduct a graph metric analysis across datasets in a given set, it was decided to scale the metrics for each dataset by the respective size of the system ($|m + n|$), allowing for comparison that isn't hindered by excessive disproportionality in the graph metric absolute values. It also allows for a fairer analysis across datasets.

This was done for each dataset. The scaled averaged metrics for the three sets of datasets are shown in Figure 8-1 and Figure 8-2. The graphs in Figure 8-1 show the relationship that the number of suppliers and consumers have on these two key graph metrics. These are results from both the **GROUP S** and **GROUP C** datasets. The graph metrics shown are scaled by the number of suppliers and consumers that make up the vertices in the graph, thereby reducing disproportionality that would make it difficult for visual analysis of the scatter plots. It also allows for a fairer analysis across datasets.

The scaled metrics were then plotted against the absolute difference between the number of suppliers and consumers in the IS network since for both **GROUP S** and **GROUP C** datasets, there exists supplier-dominant and consumer-dominant datasets. It was found that the greater the difference between the number of suppliers and consumers, the smaller the number of graph edges. Also, the greater the difference between the two sets of companies, the greater the number of graph components.

As opposed to the **GROUP S** and **GROUP C** datasets, graph metric results from the **GROUP SC** datasets, shown in Figure 8-2, were plotted against their respective total number of suppliers and consumers. It was found that, on average, a larger number of edges and a smaller number of components result from **GROUP SC** set of datasets than the other sets of datasets for the same number of companies. It was also noted that the rate of change of the graph metrics decreased as the number of suppliers and consumers increased for the **GROUP SC** set of datasets. In other words, there exists a limit to the number of connections that are possible as the number of companies increases. Reasons for this are discussed in Chapter 8.3.

<u>Plot results for **GROUP C** datasets</u> <u>Plot results for **GROUP S** datasets</u>

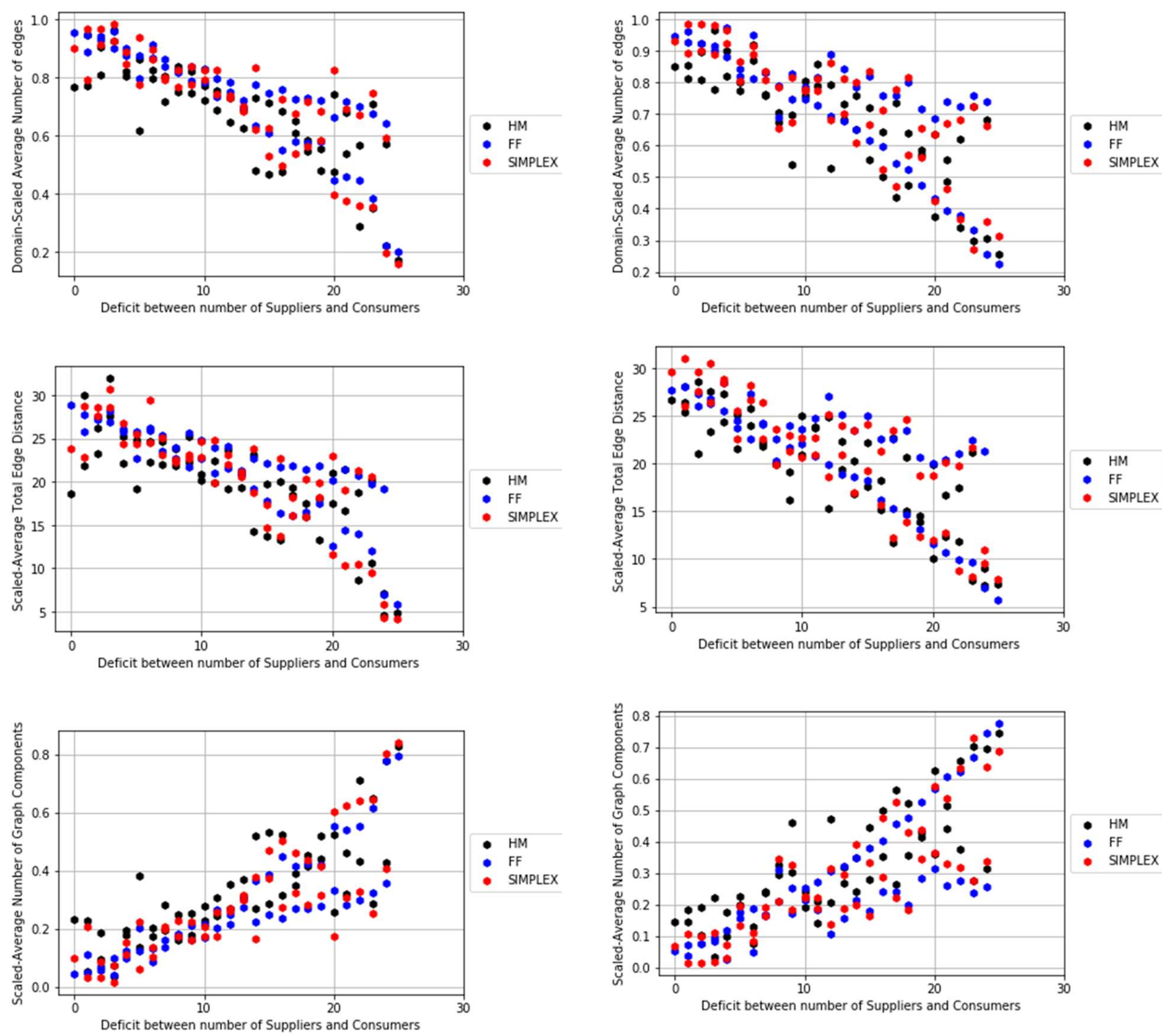


Figure 8-1:Scaled Graph metrics for the **GROUP C** (left) and **GROUP S** (right) datasets against the difference between the number of suppliers and consumers.

Plot results for **GROUP SC** datasets

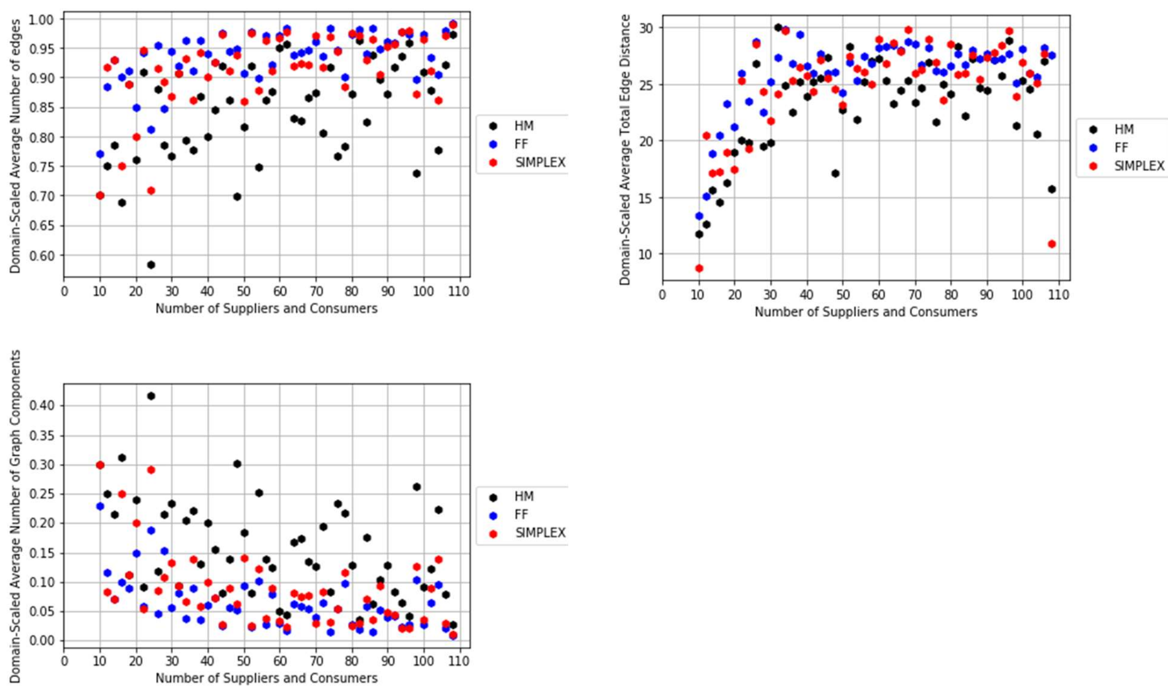


Figure 8-2:Scaled Graph metrics for the **GROUP SC** datasets against the total number of suppliers and consumers.

Algorithm comparison

As can be seen in Figure 8-2, it is difficult to conduct a visual analysis of the graphs rendered by each algorithm in the creation of IS networks with the imposed constraints. It is for this reason that, for each dataset, a Z-score normalization was conducted for the graph metric results from each algorithm. Z-score normalization is a measure of the number of standard deviations a data point is away from the mean value. For a given dataset x,the averaged graph metric X rendered from each algorithm i ($i \in [1,3]$) populates the data for Z-score normalisation. The mean $\overline{X_x}$ and the standard deviation σ_x of that data is used to calculate the Z-score value $Z_{i,x}$ of each data point $X_{i,x}$, as shown in Equation 8-1. As a result, a much clearer evaluation can be done to determine trends between the metrics from each algorithm, as is evident in the Z-score normalized results from the **GROUP SC** datasets in Figure 8-3.

$$Z_{i,x} = \frac{X_{i,x} - \overline{X_x}}{\sigma_x} \qquad \textit{Equation 8-1}$$

As seen in Figure 8-3, the HM, on average, renders IS networks with the lowest number of graph edges than the other two algorithms and the largest number of graph components, whereas the FF renders IS networks with the largest number of graph edges and the smallest number of graph components than the other two algorithms. This is interpreted from the observed Z-score values. This visual analysis allowed for the general trends to be observed and for unique graph metrics to be inferred for a given algorithm. The suspected trends are then confirmed by analysing the statistical results for all sets of datasets.

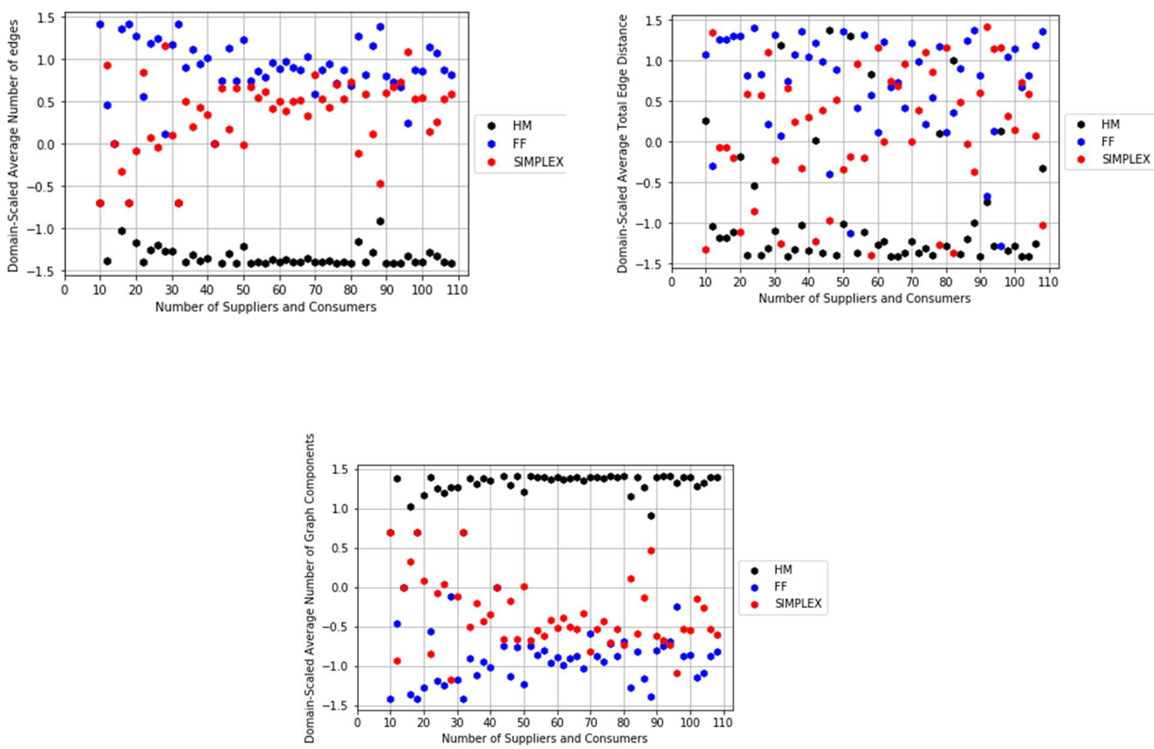


Figure 8-3:Z-score normalized graph metric results for the **GROUP SC** datasets against the number of suppliers and consumers.

8.2 Statistical graph metric results

The trends were found to be consistent with the other sets of datasets (**GROUP S** and **GROUP C**), as seen in Table 8-1, Table 8-2 and Table 8-3, where for each algorithm i, the mean Z-score normalized metric $\overline{Z_{\iota,T}}$ across a given set of dataset T (e.g. **GROUP SC**) was calculated. It must be stated again that the use of Z-score normalisation was solely to magnify the discrepancy in the metric values rendered by each algorithm to determine if unique graph characteristics could be attributed to a specific algorithm. In this case, the inclusion of the mean metric value $\overline{X_x}$ would not provide additional context for the intended analysis.

Table 8-1: Mean Z-score value for the average number of graph components.

	Mean Z-score normalized value for av. number of graph components		
Algorithm	**GROUP S**	**GROUP SC**	**GROUP C**
SM	-0.3668	-0.3511	-0.1078
HM	0.8446	1.2488	0.7919
FF	-0.4778	-0.8977	-0.6841

Table 8-2: Mean Z-score value for the average number of graph edges.

Algorithm	**Mean Z-score normalized value for av. number of edges**		
	GROUP S	**GROUP SC**	**GROUP C**
SM	0.3668	0.3511	0.1078
HM	-0.8446	-1.2488	-0.7919
FF	0.4778	0.8977	0.6841

Table 8-3: Mean Z-score value for the average total edge distance.

Algorithm	Mean Z-score normalized value for av. Total edge distance		
	GROUP S	GROUP SC	GROUP C
SM	0.2517	0.1312	0.0674
HM	-0.6000	-0.8617	-0.6175
FF	0.3482	0.7305	0.5502

In all cases, the HM produces the lowest number of graph edges and the highest number of graph components. Hence, given a group of companies, it can be said that the HM will produce a relatively smaller number of supplier-consumer pairings, which makes up an IS network and will be composed as smaller subnetworks. On the other hand, the FF algorithm will, on average, produce the greatest number of supplier-consumer pairings and the least number of graph components, meaning a denser IS network in terms of connections and a smaller number of subgraphs/clusters. It is also seen that the average total edge distance in an IS network is lowest when the HM is applied, compared to the other two algorithms, as seen in Table 8-3, where on average, the HM produced IS networks with a 9% and 6.06% lower travelling distance than SM and FF, respectively.

This puts the Hungarian method in a positive light for its application to IS since it creates networks with companies being in closer geographical proximity, which can be considered as the chief principle of IS. With the fewer number of graph edges, corresponding to a smaller number of supplier-consumer connections, it can be stated that the HM promotes efficient inter-company exchanges with respect to connections in minimising the by-product material sent to the landfill. This bias to efficient inter-company exchange is particularly important for industrial symbioses dealing with freshwater consumption (Boix et al., 2015).

On the other hand, the FF algorithm produces IS networks that are more inter-connected, with a larger number of graph edges and a smaller number of subgraphs. This means more companies are included in the IS collaboration leading to a larger number of companies gaining a competitive advantage for the same amount of demand and supply. The implications on the distribution policies are expanded upon in Chapter 8.3.3.

In terms of algorithm precision, the HM proves to be the most consistent algorithm in terms of the solution it renders for datasets of the type **GROUP SC** and **GROUP C**, since the lowest average standard deviation in the key graph metrics was found for the HM compared to the other two algorithms, as seen in Table 8-4. It was noted however that the Simplex method is significantly more consistent in the solutions it generates for datasets of **GROUP SC**, making it a more suitable algorithm for systems in which the number of suppliers and consumers are equal, in terms of solution consistency.

Table 8-4: Mean standard deviation in the graph metric results for each type of dataset and algorithm pair.

Mean Standard deviation						
Algorithm	**Num. Graph comp.**			**Num. Edges**		
	GROUP S	**GROUP SC**	**GROUP C**	**GROUP S**	**GROUP SC**	**GROUP C**
SM	0.767	0.288	0.697	0.855	0.505	0.804
HM	0.330	0.919	0.382	0.301	0.851	0.399
FF	1.353	0.795	1.404	1.334	0.945	1.385
	GROUP S 2	**GROUP SC 2**	**GROUP C 2**	**GROUP S 2**	**GROUP SC 2**	**GROUP C 2**
SM	0.641	0.259	0.561	0.662	0.429	0.640
HM	0.537	0.696	0.496	0.557	0.633	0.484
FF	1.385	0.780	1.518	1.404	0.849	1.483
	GROUP S 3	**GROUP SC 3**	**GROUP C 3**	**GROUP S 3**	**GROUP SC 3**	**GROUP C 3**
SM	0.712	0.278	0.666	0.715	0.400	0.751
HM	0.620	0.829	0.421	0.569	0.850	0.418
FF	1.375	0.806	1.448	1.376	0.891	1.448

8.3 Discussion

8.3.1 Main findings

For the case of resource distribution and transport cost minimisation, all the algorithms applied for the generation of IS networks return feasible solutions, with the minimisation of unshared goods being achieved for every dataset generated. In other words, maximal flow of by-product between suppliers and consumers is maintained, while ensuring the imposed constraints are met. However, the nature of the connections made are found to differ. The graph characteristics that each algorithm tends to render and allows for general the suitability of each algorithm to be assessed with respect to distribution policy.

It was explained in Chapter 8.1 that averaging of results, scaling and Z-score normalisation were required to analyse the results. Figure 8-4 illustrates the sequence of graph metric transformations.

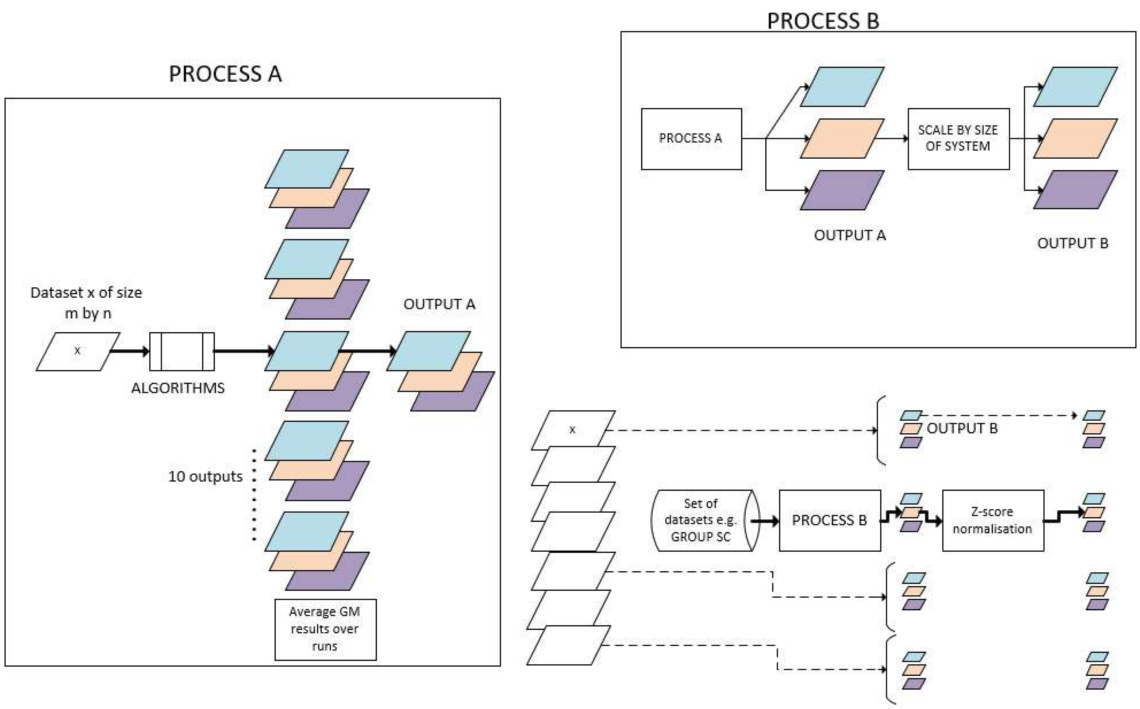


Figure 8-4: Illustration of the post-processing steps of the graph metric results.

It is seen in Figure 8-1 that one is more likely to obtain a greater number of graph edges and a smaller number of graph components when the number of suppliers and consumers are closer in value. This makes sense since when one set of companies are smaller in number, it is more likely for their total by-product requirement to be met. As such, it is not possible for any more edges to be incident to the smaller set of vertices, and this means that there will be companies, represented by the larger set of vertices (S or C), who don't take part in the IS. Since these companies are not in the matching M and don't take part in the IS, they can't be part of a subgraph or incident to an edge. Hence, the closer the number of suppliers and consumers are to each other, the more companies are part of the IS network for the given system size. As such, the number of graph components and the number of edges will be higher for the size of the system.

For the case of the **GROUP SC** datasets, the number of suppliers and consumers are equal, and a balanced bipartite graph results when an IS network is rendered by an algorithm. An interesting result was that there seemed to exist an asymptote to the scaled graph metric scatter plots in Figure 8-2. In other words, even though the absolute values of the graph metrics are not available after the scaling of the metrics, it is seen that some type of constant ratio exists between the absolute graph metric value and the size of the system. It was reasoned that this apparent asymptote is due to the bounds that were imposed on the randomized values of the variables for each company, as shown in Table 7-2. For instance, limits were imposed on the by-product amount that a company would require or have. Since these limits do not change when generating a dataset that is larger in size, they become the limiting factor to the IS network metrics, bounding the number of possible edges/connections that can be made.

All the algorithms showed the same trends. Once these general trends were understood, it became apparent that the comparison of the algorithms could commence. As seen in Figure 8-2, it is difficult to conduct a visual analysis between the results from the graph algorithms. A solution to this was to perform Z-score normalization among the graph metric results for each dataset. As seen in Figure 8-3, the Z-score normalization magnified the discrepancies in the graph metric results from the algorithms and highlighted the graph characteristics unique to the algorithms.

Figure 8-3 indicates that the HM renders IS networks with the least number of graph edges compared to FF and SM. Resultingly, it produced, on average, the greatest number of graph components. These graph characteristics are unique to HM and can be understood from an intuitive standpoint. With reference to Figure 0-6, the HM routine essentially creates one-one matchings, iteratively, until there exists no more net demand or supply, whichever value is lowest in absolute amount. Hence, given a cost matrix, traditionally termed as an *Assignment matrix,* a bijection is rendered.

As a result, the rate at which the vertex degrees and the number of graph edges increase is limited by the bijection that the Hungarian method creates to achieve a resource deficit that is equal to the difference between the total supply and the total demand. This contrasts with the FF and SM, which tend to be biased to many-to-one matchings. Therefore, a distinction can be created for the HM by stating the class of matching algorithms it belongs to, which can be defined as **Bijective-Bipartite Matching Algorithms (BBMA).**

Furthermore, the values in the Assignment matrix are computed by a function that prioritizes matchings between a supplier and consumer, who are not only in closer geographical proximity or have similar resource unit price allocations, but also result in the minimum amount of unshared goods left over/required after the matching, as seen in Equation 5-16. Hence, “exact matchings” are promoted, thereby reducing the possibility of additional edges between either company and other companies. This type of policy is not upheld in the Simplex method or Edmonds-Karp algorithm, even though the minimisation of unshared goods is a constraint. This situation is identified as a *Lack of guidance in the matching* for the FF and SM and is elaborated on below.

Lack of guidance in matching for the Edmonds-Karp algorithm

The Edmonds-Karp algorithm is an implementation of the Ford-Fulkerson method (Cormen et al., 2009), with the difference between the two being that the Edmonds-Karp algorithm implements the Breadth-First search (BFS) algorithm for choosing an augmenting path from a source node to a sink node, whereas the augmenting path chosen in the Ford-Fulkerson method is an arbitrary choice. The BFS algorithm acts as a shortest path algorithm, finding the path from source to sink that contains the minimum number of edges.

If there is more than one path from source to sink having the minimum number of edges, then the path chosen is that path that contains the right-most edge from the root vertex, as seen in Figure 8-5 (the red path will be chosen). The problem with the Edmonds-Karp algorithm is that its advantage over the Ford-Fulkerson method is rendered ineffective with the current graph representation since all paths from the “super-source” node to the “super-sink” node are of the same length of 3 edges. In such a case, the choice of the augmenting path is biased to the right-most edge from the “super-source” node.

As a result, if an exact matching exists, in most cases, it will not get chosen since the chance of the matching being made is dependent on the position of the companies in the graph, rather than their resource amount values. A solution to this issue would be to alter the BFS algorithm by representing the shortest paths as those that promote exact matchings. This would result in an augmenting path being chosen that connects a supplier vertex to a consumer vertex whose resource allocation is the most similar in absolute value.

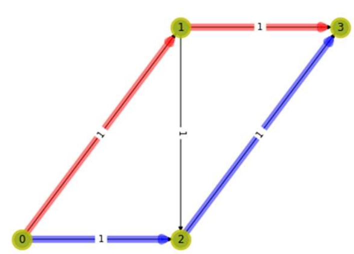


Figure 8-5: An example of a tie in the chosen augmenting path in the BFS routine for the Edmonds-Karp algorithm.

Lack of guidance in the matching for the Simplex method

For the case of the Simplex method, it is the transportation problem formulation that not only introduces bias to the Simplex algorithm but also does not promote exact matchings, resulting in an IS network that has more edges than an IS network in which more exact matchings are made. Consider the transportation problem formulation. The functional constraints ensure that no supplier can supply a resource amount more than it has, and no consumer can accept a resource amount more than it demands. The coefficients in the objective function are computed using Equation 5-5, ensuring that when the Simplex algorithm is called, the most negative coefficient is chosen, and the respective variable becomes the entering basic variable (EBV).

In other words, the variable that enters the basis (entering basic variable) corresponds to a matching between a supplier and consumer that have the closest geographic proximity and have similar unit price allocations for the resource. However, since there may be cases in which the total supply is not equal to the total demand, the Two-phase Simplex method has to be incorporated, meaning that the inclusion of the objective function coefficients that reflect the policies are only done in Phase 2. Consider a simple 2 supplier and 2 consumer system. For Phase 1 of the 2-phase Simplex method, the initial structure of the Simplex table can be generalized and is shown in Figure 8-6.

	Z	x_{11}	x_{12}	x_{21}	x_{22}	x_{a1}	x_{a2}	x_{a3}	x_{a4}	RHS
Z	1	k_{11}	k_{12}	k_{21}	k_{22}	0	0	0	0	0
x_{a1}	0	1	1			1				θ_{S_1}
x_{a2}	0			1	1		1			θ_{S_2}
x_{a3}	0	1		1				1		$abs(\theta_{C_1})$
x_{a4}	0		1		1				1	$abs(\theta_{C_2})$

Figure 8-6: General structure of the Simplex table (ST) for Phase 1 of the Simplex method.

Phase 1 requires that $Z = f(x_{ai})$, the artificial variables, as seen Figure 0-2. When this is done, the first row of the ST must then not have any entries for the artificial variables, since they are the basis (Hillier and Lieberman, 2010). This is done by performing row operations with the other rows in the ST. The resulting table is as follows:

	Z	x_{11}	x_{12}	x_{21}	x_{22}	x_{a1}	x_{a2}	x_{a3}	x_{a4}	RHS
Z	1	-2	-2	-2	-2	0	0	0	0	0
x_{a1}	0	1	1			1				θ_{S_1}
x_{a2}	0			1	1		1			θ_{S_2}
x_{a3}	0	1		1				1		$abs(\theta_{C_1})$
x_{a4}	0		1		1				1	$abs(\theta_{C_2})$

As a result, all the coefficients for the decision variables in the objective function are identical, and this results in a tie for the entering basic variable. In situations like this, the tie is broken arbitrarily, with the first column from the left being chosen. Clearly, x_{11} will always be chosen initially in phase 1 of the 2-phase simplex method as the entering basic variable. Hence, from the start, there is no approach for choosing the EBV in a way that prioritizes exact matchings, resulting in an initial basic solution being produced at the end of Phase 1 of the 2-phase simplex method that is, in most cases, an IS network that has more edges than required to meet the constraints.

Therefore, it can be concluded that the strength of the HM lies in its bias to creating matchings in which the resource deficit between the supplier and consumer pair is a minimum, making it the most suitable algorithm that has been applied at transportation cost minimisation. It is suspected that it is because of this bias to creating exact matchings, which is not present in the other algorithms, that the HM is more consistent in terms of the solutions it generates, as seen in Table 8-4.

It was intended to confirm these observations concerning the lack of guidance in the FF and SM, deducing a solution by empirically determining the average time complexity for the algorithms. It was anticipated that the average time complexity for FF and SM would closely approach their respective worse case time complexity, indicating that, on average, the lack of guidance imposed by the graph representations results in the suboptimal intermediate pairings being made. These pairings would have to be revoked since compatible pairings contribute to a matching M. Compatible pairing are those pairings that do not violate any of the constraints imposed on the respective LP of the network flow problem. The flow continuity constraint and capacity constraints are the LP constraints for the maximal flow problem, and the LP constraints for the transportation problem are its functional constraints.

8.3.2 Determination of Empirical Time complexity

A simulation was run where 32 **GROUP SC** datasets were created, with the i^{th} dataset containing an equal number of suppliers and consumers, resulting in the total number of companies/graph vertices being $2 \times (10 \times i)$. This meant that the largest dataset would have 320 suppliers and 320 consumers, each with randomized values for their product attributes. The algorithms processed each dataset once, instead of 10 times to save on computation time, and 32 datasets were processed instead of a planned 50 **GROUP SC** datasets, as processing the 33rd dataset for the SM took well over 28 hours and still did not come to completion. Consequently, it was decided to terminate the simulation and record the results of the first 32 datasets.

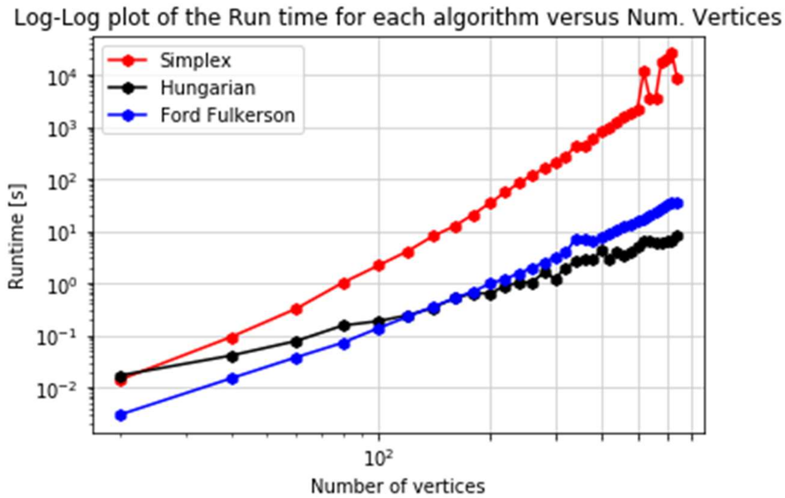


Figure 8-7: Graph showing the runs to determine the time complexity of each algorithm.

The simulation was conducted to determine the complexity function of each algorithm as a function of the number of graph vertices V. A log-log plot was used to observe the results. It is seen that the Simplex algorithm does take significantly longer to process larger datasets than the other two algorithms. While the FF algorithm does perform marginally the best in terms of time complexity for smaller datasets, for larger datasets, it is the HM that performs the fastest at determining IS networks. The empirical average time complexity of HM and FF was found to be $O(V^2)$ and $O(V^3)$, respectively, since the gradient of the black (HM) and blue (FF) linear functions in Figure 8-7 were 1.88 and 2.81, respectively.

It is reasoned that the average time complexity for FF is due to the fact it will take $O(E)$ time for the BFS to find the shortest augmenting path in the flow network. FF runs so long as there is an augmenting path from the super-source to the super-sink vertex, which is the condition for a while loop in the FF routine. Given that the bottleneck edge from the augmenting path must be determined in a given loop, and there can be at most $E \times V$ bottleneck edges in the residual graph G_f the time complexity is calculated as $O(E \times V \times E)$, with $E = V$ in the residual graph, resulting in $O(V^3)$.

Similar reasoning applies for the HM where there are at most E edges, hence there are only E iterations to find a good path. Since there are at most E price updation steps required to find a good path and that the BFS tree can be computed in $O(E)$ time, the worst-case time complexity is $O(V \times V \times E)$. Since $E = \frac{V}{2}$, the time complexity is $O(V^3)$, which is different than the average time complexity empirically determined for the HM of $O(1.88)$. This highlights the fact that the HM finds the solution without needing to enumerate all the suboptimal pairings to render a bijection.

On the other hand, the FF approaches the worst-case time complexity since all paths have the same length and, therefore, approaches the worst-case time complexity. The red graph in the log-log plot in Figure 8-7 shows that there is a slight change in gradient as the number of vertices increase, indicating that there may be an exponential dependence that the time complexity has on the number of graph vertices, which seems to confirm that the time complexity of the SM is exponential.

To determine whether this is true, a regression was done on the Time complexity versus the number of graph vertices for the SM data points. As can be seen in Figure 8-7, the last few data points for the SM don't follow the same trajectory as the rest of the points, with some datasets taking one order higher than the previous and next dataset. This may be due to the increased computer usage at the time (Intel ® Core ™ i7-10510U). Hence, these points were omitted when performing the regression. It was found that the exponential function $y = 7.4552(1.011)^V$ fit the time complexity data for the SM with a reasonably high R^2 value of 0.993. This concluded that the Simplex method could be regarded as an exponential time complexity algorithm.

Table 8-5: Algorithm time complexity (seconds) for each type of dataset.

Algorithm	**Total Time complexity (s)**		
	SICC	SICI	SCCI
SM	320.94	577.38	367.81
HM	207.41	249.28	190.18
FF	158.73	232.86	158.87

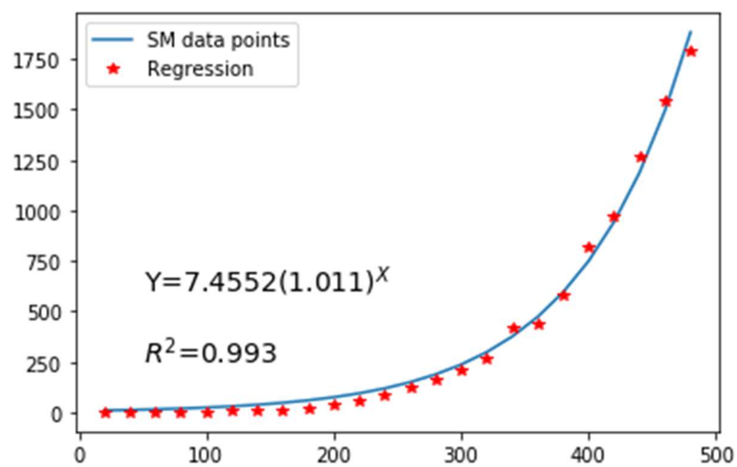


Figure 8-8: Regression on the SM time complexity datapoints, showing an exponential trend.

In terms of the rigorous study, the Edmonds-Karp algorithm proves to be, marginally, the more computationally efficient algorithm, being 5.83% faster on average at determining IS networks for the different types of datasets than the Hungarian method, while the Simplex method is the most computationally expensive, taking 43.34% longer, on average to determine IS networks than the Edmonds-Karp algorithm, as seen in Table 8-5. It was reasoned that the recomputing of the edge capacities as shown in Figure 0-4 (represented by its respective graph weight matrix) after a supplier-consumer pair is made takes a smaller amount of time complexity than if the flow network was solved in the traditional way that the FF method proceeds in (Figure 0-3).

While the weighting of travelling distance and fair resource distribution, based on resource unit price, complicated the FF application and required alteration in the algorithm's subroutines, the FF still produces feasible IS networks faster. However, it is noted that in practical application, where the number of suppliers and consumers may not be a large number, algorithm time complexity may not be a factor, since real-time matching may not be important given the evidence in the literature that social factors (such as participant satisfaction) may require review of the suggested IS network by the appropriate stakeholders (Boix et al., 2015, Boons and Baas, 1997, Kastner et al., 2015).

Table 8-6: Algorithm time complexity (seconds) for different runs of each type of dataset.

Algorithm	**Time complexity (s)**		
	SICC2	SICI2	SCCI2
SM	385.41	744.95	454.71
HM	259.70	336.56	254.23
FF	191.30	306.41	194.93
	SICC3	SICI3	SCCI3
SM	382.84	821.94	436.84
HM	275.37	327.01	253.90
FF	196.92	319.36	193.41

8.3.3 Implications to Distribution policy

It was found that the HM is the only algorithm among the three whose solution strategy is guided by the distribution policies imposed. The FF algorithm uses a BFS method to obtain the augmenting path, which initiates a pairing between a supplier and consumer. However, the graph representation of the IS problem has a constant path length of 3 edges. As such, all augmenting paths are non-inferior in the light of the BFS algorithm. This results in the order of matchings being made by some default, as shown in Figure 8-5.

For the 2-phase SM, a similar thing occurs, where in Phase 1, all the k_{ij} coefficients have a value of -2. Due to this, the Simplex algorithm during Phase 1 will see all entering basic variables as non-inferior candidates, and a default choice will be made. Nonetheless, the FF and the SM algorithm still display bias to certain policies and is explained below.

Economically biased distribution policy (ECD)

This distribution policy was defined in Chapter 4 as one with a bias to increasing economic gain. An example of this type of policy is the minimisation of transport costs, which will result in increased economic gain experienced by the suppliers and consumers. With respect to this policy, the HM is superior to the other algorithms since it produces IS networks with a smaller number of edges and a smaller travelling distance. As a result, the cost of transportation shared by the supplier-consumer pair will be lower, allowing for a larger profit for the companies.

Environmentally biased distribution policy (EVD)

This policy defines a distribution that is concerned with a reduction in environmental impact. An example of this is minimising greenhouse gas emissions associated with the transportation of the exchanged material. This distribution would focus on pairing companies that are in closer geographical proximity. Another example of this policy is in the minimisation of by-product material not shared between a supplier-consumer pairing. In this way, suppliers would have their waste by-product distributed such that their environmental impact from waste disposal and greenhouse gas emission is minimised. Hence, the HM is a superior algorithm compared to the FF and SM. The bijection that is created at each iterative application of the HM to the IS problem results in closely matched suppliers and consumers with respect to their σ values being paired. In this way, "exact matchings" are created, which results in the smallest deficit in unshared by-product material. Companies that are geographically closer are also favoured by the HM, which is why the algorithm rendered IS networks with the smallest travelling distance.

Social construction-based distribution policy (SCD)

While the EVD policy is concerned primarily with meeting increased environmental performance, and the ECD policy is primarily concerned with optimising cost savings, the SCD policy is unique, as it is focused on the social aspects of the IS network.

Trust

Trust between companies has been noted to be an important precursor for the progression of IS networks (Hewes and Lyons, 2008). When companies work together to collectively reap the benefits of IS in a fair manner, a higher collective payoff is achievable than if companies focused on solely optimising their own objective. This notion was confirmed by Chew et al. (Chew et al., 2009) when they applied game theory to model the interactions of participants in an EIP and found that a higher collective payoff was achievable when a cooperative scheme was used as opposed to the non-cooperative scheme. Furthermore, the collapse of an IS network is most likely when competitive behaviour is prevalent rather than a cooperative one and has been termed by Yazdanpanah et al. (2020) as the *Industrial symbiosis Implementation problem*.

As such, an important distribution policy is one based on the social construct of trust. According to Lewis and Weigert (1985, as cited in Doménech and Davies, 2009) found that the size of a network can negatively impact the trust in the network, with smaller networks, like Kalundborg, better suited for the development of trust than larger networks (Doménech and Davies, 2009). Hence, IS networks rendered from the HM are more likely to have long-term longevity than those rendered from the FF method since the HM produces a greater number of graph components than the FF. In other words, a larger number of sub-networks/clusters made of a smaller number of companies result from the HM than the other two algorithms.

Fair distribution

An SCD policy may be sought that is based on "fairness". One can interpret fairness in 2 ways.

1. A distribution based on fairness may seek to include as many companies as possible since economic benefit is derived from being a part of an IS.
2. A distribution that pairs suppliers and consumers who derive similar economic benefit from the transaction or results in an IS network in which the majority of companies have a similar level of influence/importance.

Point 1 implies that the distribution would seek to include as many suppliers (sources) and consumers (sinks) for a by-product material as possible. Given that there is a total supply amount by m companies and demand amount n companies, the distribution would seek to include as many of the m companies and n companies to be in the IS such that the difference in supply and demand ($|\sigma_S - \sigma_C|$) is met. In other words, if there is a net supply left after the demand has been met, this surplus of by-product material would be due to the sum of surplus material from as many suppliers as possible, thereby maximising the number of supplier companies that gain some economic and environmental benefit from the IS.

The FF and SM excel in creating IS networks with this type of distribution policy compared to HM. This is especially true for FF, which showed that it creates IS networks with the largest number of edges (connections) between suppliers and consumers. As a result, these algorithms are good at maximising the number of companies that gain a competitive advantage in an IS network. However, with this increased number of connections comes an increase in travelling distance, as was noted in Table 8-3 for the FF, which would result in higher transportation costs and greenhouse gas emissions.

Point 2 can be achieved by pairing suppliers and consumers whose economic value α (buying/selling unit price) attributed to the by-product material is most similar. Furthermore, one could distribute the by-product supply in such a way that the importance of companies to the network or their level of influence is as similar as possible. A way to measure this is using the Eigenvector Centrality (EVC) measure (Wasserman and Faust, 1994). Node (Vertex) centrality is an important concept in SNA and has been used to determine the importance of a node due to the topology of the network/graph (Borgatti and Li, 2009). Each vertex in the graph can be assigned an EVC value, ranging from 0 to 1. A higher value indicates that the vertex has a stronger influence than a vertex with a lower value. In this way, EVC can rank vertices by their importance, where the importance of a vertex is evaluated based on the number of connections it has to other important nodes. Where an IS network includes many subgraphs (number of graph components>1), the EVC for a vertex would be with respect to its subgraph, and not the IS network as a whole.

A fair distribution may be considered as one where all of the companies in the network have a similar degree of influence/importance; there is no power imbalance. In other words, there is no hierarchical nature to the IS in which power dynamics may ensue. Power imbalances are a barrier to IS implementation (Mulrow et al., 2017) or could lead to network instability. Fair distribution in this light is best achieved by the HM compared to the other algorithms. Consider a simple 3 by 3 system shown in Figure 8-9 :

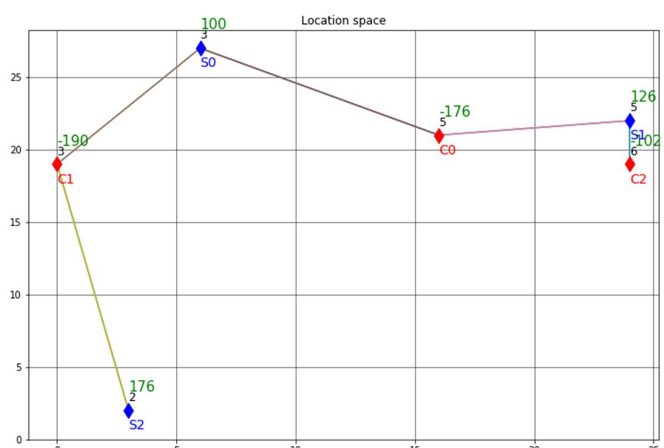


Figure 8-9: Simple IS system for EVC demonstration.

The MST graph is shown for the 3 supplier-3 consumer IS problem in Figure 8-9. When each algorithm is applied to the problem, three different IS networks result as shown in Figure 8-10.

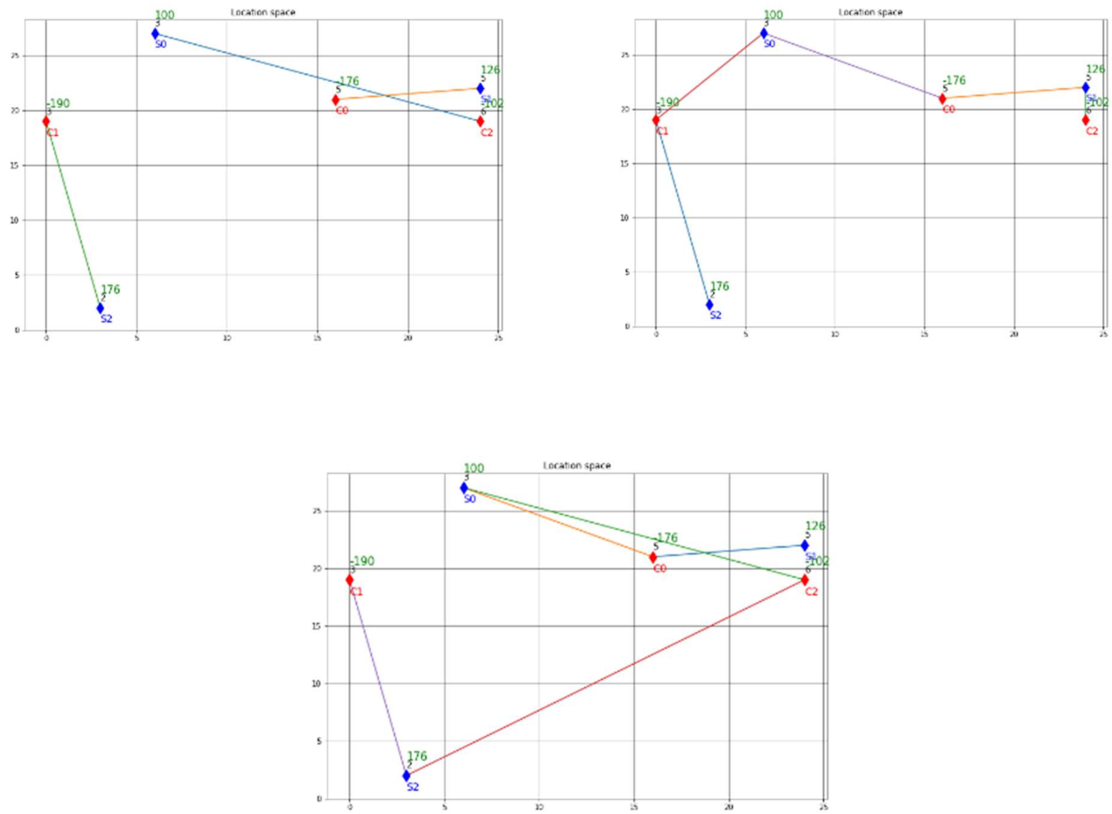


Figure 8-10: IS networks for the HM (left), SM (right) and FF (centre) with different EVC profiles.

Table 8-7: Graph metric results for the IS networks rendered by each algorithm.

Metrics	HM	FF	SM
Num. edges	3	5	5
Num. Graph Components	3	1	1
Total travelling distance (units)	96.09	203.36	99.97
E.V Degree centrality	0.408	Suppliers:[0.52, 0.23,0.41] Consumers: [0.41,0.23, 0.52]	Suppliers: [0.52,0.41,0.23] Consumers: [0.52,0.41,0.23]

Table 8-7 shows the graph metric results for each graph produced by the respective algorithm. It is seen in the case of FF and SM, there exist different EVC values for the supplier vertices and consumer vertices, with Supplier S_0 having a higher level of importance in the IS networks for the FF and SM (EV=0.52). However, for the HM, all vertices have the same EVC value (EVC=0.408). In this case, it is because each supplier is paired to one consumer. In general, one may say that the HM results in IS networks that are fairer because of the characteristic lower number of graph edges it produces and the bias it is able to afford to “exact-matchings”.

Hence, even though the EVC values were not calculated for each IS network deduced in the rigorous study presented in Chapter 7, it is anticipated that there will be a smaller discrepancy in the EVC values for the vertices of an IS network by the HM than the FF or SM because of the smaller number of graph edges and a larger number of graph components it produces.

If this is the case, then the HM is a more useful matching algorithm than FF and SM since it excels in all outlined aspects of the SCD policy (trust and fairness). The purpose of the methodology in Chapter 7 can be summarised as determining whether graph metrics can be used to differentiate between the results from each of the algorithms and explore how they can be related to distribution policies, outlined in Chapter 4.

With the completion of the rigorous study, together with its analysis and discussion, it is clear that there are distinctions in the nature of the graphs obtained by the three algorithms. The HM deduces IS networks with unique graph characteristics, as does the FF algorithm, with both algorithms deemed to favour different distribution policies. A connection was made between the graph metrics and the implications on the distribution policy using the results of the study. Furthermore, it was confirmed that the intelligence of the FF in choosing an augmenting path by exploiting the BFS algorithm is rendered ineffective due to the graph representation. Similarly, due to the same weighting given to the objective function coefficients in Phase 1 of the 2-Phase simplex method, the efficiency of the SM in deducing an optimal IS network is diminished. With the positive distribution policy implications for the HM, inferred from the graph metrics, it was determined that the HM is the superior algorithm in deducing IS networks to the SM and FF algorithm.

To determine if the graph metric analysis is a suitable approach in ranking the algorithms with respect to distribution policies and to confirm the superiority in IS network creation by the HM, Chapter 9 presents a case study that quantifies the benefits gained with respect to the distribution policies from the IS network that each of the algorithms produces. Specifically, the minimisation of transportation costs will be evaluated, and the by-product distribution with respect to the ECD, EVD, and, most especially, the SCD policy for creating fair networks that nurture trust. Eigenvector centrality will be further elaborated upon and will be used to quantify the fairness of the network.

9 CASE STUDY: APPLICATION OF GRAPH-BASED ALGORITHMS FOR INDUSTRIAL SYMBIOSIS NETWORK CREATION

A case study is now developed to further explore the influence of the matching algorithm choice on the character of distribution achieved. Two statements were made in Chapter 6 surrounding the creation of feasible IS networks and reiterated as follows:

Statement 1: For a given assignment matrix $\underline{\acute{A}}: \acute{A_{ij}} = f(PA_{Si}, PA_{Cj})$*, the rendered IS network is dependent on the algorithm choice.*

Statement 2: For a given algorithm, the order of each company's product attributes in its respective product attributes list affects the IS network that is rendered.

These statements were confirmed in Chapter 7 and 8 where the methodology was carried out, and the results were analysed, respectively. Graph metrics were found to be a good analytical tool in identifying unique graph characteristics rendered from the different algorithms. This was then linked to the impact that it could have with respect to different distribution policies. It is now intended to evaluate these findings with respect to their policy implications

Three policies were defined in this study and are listed as follows:

- Environmentally biased Distribution policy (EVD)
- Economically biased Distribution policy (ECD)
- Social construction-based Distribution policy (SCD)

It is possible to rigorously compare the algorithms by reducing the characteristic graph metrics to a single metric/index, which can benchmark performance. The composition of this metric is based on the cost associated with a particular matching. Note that this cost comprises environmental impact as well as economic competitiveness and that the relative weights of these individual costs are as determined by the distribution policies.

Therefore, this chapter is dedicated to outlining a possible method of quantifying distribution policies. Alongside this will be a case study that will test whether the previous observations surrounding IS network creation by the algorithms is consistent. Slight modifications to the problem statements are made to challenge the observations gathered in Chapter 8 regarding the algorithms' performances for IS network creation. For instance, the payload of a distribution vehicle will be accounted for in the exchange of waste by-product material. Furthermore, the size of the IS case study will be smaller than the datasets in the rigorous study from Chapter 8, with the number of suppliers and consumers being smaller and closer in number, which will only amplify the discrepancies in the weightings of connections given that there are fewer options to choose from.

9.1 Case Study: Plastic waste distribution

Plastic is a highly versatile material that is used in several industries such as the food industry, electronic, and construction industry (Alabi et al., 2019). It is said to be durable, chemically inert, inexpensive, lightweight, and easily processed (Seay et al., 2020), which explains its popularity in various industries (Singh and Sharma, 2016). However, despite these many benefits, plastic waste still causes great environmental damage in a multi-faceted way (Yogalakshmi and Singh, 2020, Verma et al., 2016). Plastics are high molecular mass compounds that usually contain other toxic substances (Verma et al., 2016). The waste mismanagement of plastic causes concern since the possibility of toxic contaminants leaching into the soil and subsoil affects soil quality and can affect marine life near water bodies (Verma et al., 2016). Waste disposal methods such as landfilling, incineration leads to the release of greenhouse gases and emission of toxic substances (Verma et al., 2016).

While there are over 300 recycling companies in South Africa, only 46.3% of plastics were recycled in 2018 (Plastics SA, 2019). A promising approach to reduce plastic waste is revalorisation of waste plastic via chemical recycling (Dai et al., 2021, Rahimi and García, 2017). Chemical recycling is a process where the plastic is broken down into its monomers so that it can be used to create new plastic materials or products (Rahimi and García, 2017). The multi-national oil and gas company, Shell, is using pyrolysis to convert plastic to liquids that can be made into useful chemicals (Casparie, 2020).

This case study is between a set of companies that seek to convert waste plastic to new plastic materials, chemicals, and fuels, and a set of companies that need to discard their waste plastic. It is sought to develop an IS network that is optimal with respect to the distribution policies. The geographical position of each company is shown in Figure 9-1, with the black edges representing the roads and the vertices representing the site locations. The companies are situated within a $2500km^2$ area. Table 9-1 and Table 9-2 show the product attributes for the supplier and consumer companies, respectively. The first step is in determining the shortest travelling distance for each supplier-consumer pair. Dijkstra's algorithm is used for this purpose and in this way, the matrix $\underline{D}$ is created.

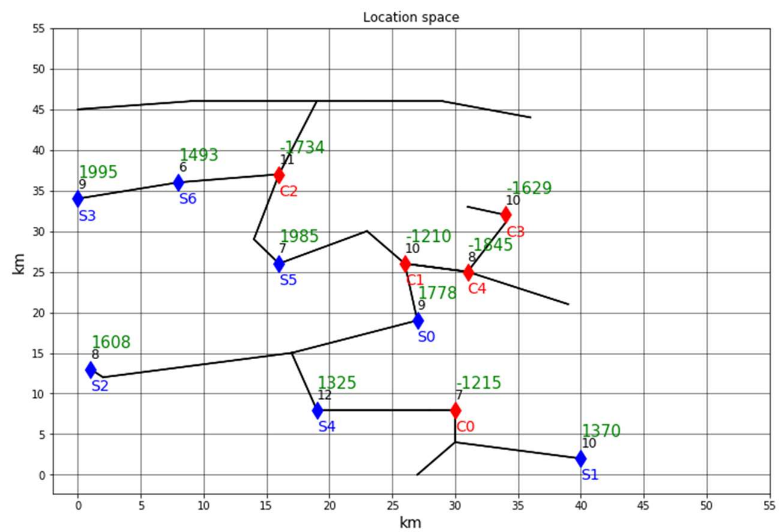


Figure 9-1: Geographical location of the suppliers and consumers of plastic waste.

Table 9-1: Product attributes for suppliers of plastic waste.

Product attributes	S_0	S_1	S_2	S_3	S_4	S_5	S_6
Resource capacity σ (kg)	1778	1370	1608	1995	1325	1985	1493
Unit Price α (R/kg)	9	10	8	9	12	7	6
Geographical location (θ_x, θ_y)	(27,19)	(40,2)	(1,13)	(0,34)	(19,8)	(16,26)	(8,36)

Table 9-2: Product attributes for consumers of plastic waste.

Product attributes	C_0	C_1	C_2	C_3	C_4
Resource capacity σ (kg)	-1215	-1210	-1734	-1629	-1845
Unit Price α (R/kg)	7	10	11	10	8
Geographical location (θ_x, θ_y)	(30,8)	(26,26)	(16,37)	(34,32)	(31,25)

Before defining the cost functions that would weight the graph edges for each algorithm, one must first develop a way to quantify the alignment of a particular matching with the defined distribution policies.

9.2 Modelling Distribution policies

Regulatory policies are the main drivers for companies adopting more sustainable supply chains (Darnall et al., 2019). When companies participating in a supply chain are distributed geographically, the company exchanges become dependent on the regional policy and local practices (Manning et al., 2012; Darnall et al., 2008; Montabon et al., 2016 as cited in Darnall et al., 2019). The regulatory policy can be in the form of traditional regulations (in the form of bans), market-based regulations (incentivised regulations), and involuntary-based regulations (to induce voluntary agreements to reduce environmental impact), all of which is defined and outlined by Darnall et al. (2019).

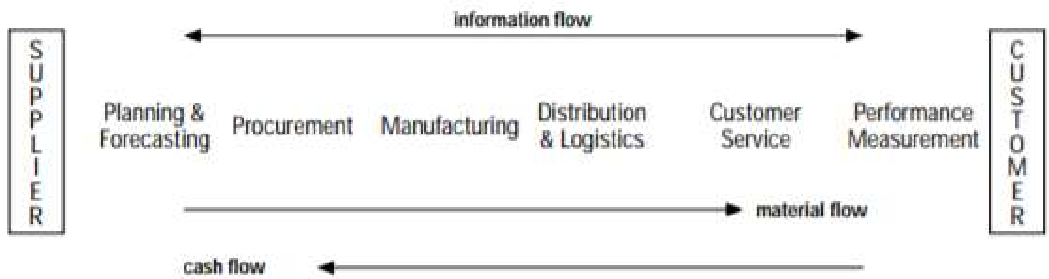


Figure 9-2: Example of a supply chain for a manufacturing company, extracted from Spekman et al. (1998).

However, whatever the form of the regulatory policies, they affect different stages of the supply chain, an example of which is shown in Figure 9-2. At each stage, the choice of the suppliers is determined (Darnall et al., 2019), thereby affecting the distribution of the commodity in question. Therefore, it may be more appropriate to refer to these policies as distribution policies, as they dictate the supplier-consumer partnerships created. Of specific interest is the distribution policies in the distribution and logistics stage of the supply chain in Figure 9-2.

It is possible to determine the cost of a distribution policy choice by way of an index to speak to the value of a matching. This index/cost can be defined as a linear combination of cost functions of the relevant features of the matching. For example, the index I for distribution policy k can be defined as Equation 5-2. The variable x_i defines a feature of the matching, of which there may be T features. A function f_j defines a specific cost that has been identified as a function of b features and the coefficient ${\psi_j}^k$ corresponds to the weighting that the distribution policy gives to each type of cost.

$$I_k = \sum_{j=1}^{T} {\psi_j}^k f_j(x_{1,} x_2 \dots x_b)$$

Equation 9-1

Different policies will have different values for the weighting coefficients. Hence, by representing Equation 9-1 in vector form (as Equation 5-2), a distribution policy k can be characterised by the vector $\underline{\psi}^k$, since it is a unique identifier of the distribution policy and $\underline{\psi}^k$ represents the weighting the distribution policy k gives to the cost vector of matching features.

$$I_k = \underline{\psi}^k . \underline{f} \qquad \textit{Equation 9-2}$$

For this study, three distribution policies were defined, which are the Economically biased distribution policy (ECD**)**, Environmentally biased distribution policy (EVD), and the Social construction-based distribution policy (SCD). While there are several features that one can choose from, four features were chosen for this study, which are:

1. x_1: The number of connections (represented by the number of graph edges)
2. x_2: Mass of CO_2 emitted (kg)
3. x_3: Distance travelled (km)
4. x_4: Number of important companies in the network

As was alluded to in chapter 8, the importance of a company is the degree of influence it has in the network and can be determined by the eigen vector centrality graph metric. This will be elaborated upon later in the chapter. A choice was made for the index to be the monetary cost since it is possible for each feature x_i to be converted to a cost using some function f_i. What follows is the process of converting each feature to a cost.

Distance travelled converted to transportation cost

Transportation costs are a significant barrier to IS implementation (Doménech and Davies, 2009). Hence, it is best to seek IS networks that result in lower travelling distances, which can be used to indicate the size of the network. A convenient measure for the transportation costs is to estimate fuel consumption for the journey, which accounts for the majority of the logistics cost (Havenga, 2010). One can estimate the fuel consumption by assuming that the fuel economy FE (Distance travelled per litre of fuel) and the price of fuel per litre POF (Fuel price) are constant values.

Hence, given a distanced travelled (x_3), the associated transportation cost ($\boldsymbol{f_1}(\boldsymbol{x_3})$) in Rands is as shown in *Equation 9-3*:

$$Transp.Cost = x_3 \times POF \frac{R}{Litre} \times FE \frac{Litres}{100km} \qquad \textit{Equation 9-3}$$

An estimate of the average fuel economy for a delivery truck (single unit 2 axle 6 tire or more) was found to be 6.4 gallons of gasoline-equivalent/mile (Federal Highway Administration, 2018), or 36.752 litres of gasoline-equivalent/100km and the price of gasoline was found to be R16.15 per litre (BusinessTech, 2021).

Mass of CO_2 emissions as an emissions tax

To quantify the environmental impact of the IS, the carbon dioxide emissions from transportation can be determined. The CO_2 emissions can be calculated by first working out the amount of fuel consumed for a given distance travelled x_3 (km) using the fuel economy FE $(\frac{L}{100km})$. Based on an estimate of the GHG_{CO_2}, the average amount of CO_2 per L of gasoline, one can determine the mass amount of CO_2 released per km travelled. For gasoline, the factor GHG_{CO_2} was taken to be $2.8kg\frac{CO_2}{litre}$ (Cefic; ECTA, 2011). Hence, the mass of CO_2 emitted is:

$$Mass\ CO_2\ emitted = x_3 \times FE\frac{Litres}{100km} \times GHG_{CO_2} \qquad \textit{Equation 9-4}$$

Based on the calculated amount of CO_2 emissions, one can estimate the tax incurred from CO_2 emissions, which, according to Van Heerden et al. (2016), has been set at R120 per ton of CO_2 emitted by the South Africa National Treasury.

Hence the fuel tax can be calculated as follows:

$$Fuel\ tax = \frac{R120}{1000kg} \times x_2 \qquad \textit{Equation 9-5}$$

In this case, x_2 is the mass of CO_2 emissions and $\boldsymbol{f_2(x_2, x_3)}$ is the $Fuel\ tax$.

The number of connections

The number of connections in the IS network speaks to the network complexity of the IS network. It was noted by Nobel and Allen (2000) and Aviso et al. (2011) that excess network complexity is a problem for inter-company exchanges and can be considered as an investment cost (Boix et al., 2015). For the purposes of this study, it is assumed that the investment cost is proportional to the number of connections (x_1) in the network and the total travelling distance (x_3) where, given a standard investment cost per unit distance $(\frac{R}{pairing.km})$ for a single supplier-consumer pairing (IC), the total investment cost $(\boldsymbol{f_3(x_1, x_3)})$as a result of network complexity is:

$$Total\ investment\ cost = IC \times x_1 \times x_3 \qquad \textit{Equation 9-6}$$

The number of important companies in the network

As was alluded to in Chapter 8, the Eigenvector centrality (EVC) of a vertex is a measure of the level of importance or influence of the vertex in the network. Values of Eigenvector centrality range from 0 to 1, with a higher value implying higher importance of the vertex in the network. Given a matching M, the EVC of each vertex can be determined. A vertex that has a higher EVC value would not only indicate that it is important in the network but that it is connected to other vertices of high importance. If an IS network is made up of subgraphs, where the number of graph components is greater than 1, then the EVC value of a vertex would be determined with respect to its subgraph and not the IS network as a whole.

With this understanding, one can postulate that networks that have a smaller EVC standard deviation may be considered as a fairer network than those with a larger value. This would imply that there are more vertices that have similar EVC values and, hence, a more equal level of influence in the network. This would go a long way in terms of network stability since power imbalances are a barrier to IS implementation (Mulrow et al., 2017) and can lead to network failure if companies with a high level of importance leave the network.

Since the EVC metric expresses a vertex's importance based on which other important vertices it is connected to, the EVC value of a vertex is dependent on that of the other vertices it is connected to. Hence, given the adjacency matrix $\underline{A}$ of a graph G, which stores all the connections of the graph, the EVC value γ of a vertex i can be determined as shown in Equation 9-7.

$$\gamma_i = \sum_j A_{ij}\gamma_j\,, \forall\, j \neq i \qquad \textit{Equation 9-7}$$

Equation 9-7 shows the dependence of a vertex i's EVC value on the EVC values of all vertices it shares a graph edge with. Equation 9-7 indicates that the EVC values for all vertices connected to vertex i to be known for γ_i to be calculated. This can be determined recursively, where the EVC values for all vertices, that are not vertex i, are initialised as 1, and would allow for γ_i to be determined. This is shown in Equation 9-8.

$$\gamma_i(t+1) = \sum_j A_{ij}\gamma_j(t)\,, \forall\, j \neq i \qquad \textit{Equation 9-8}$$

Equation 9-8 is computed recursively until an equilibrium is reached at the n^{th} iteration, where $\gamma_i(t+n) - \gamma_i(t+n-1) \ll 0$. Theoretically, as $t \to \infty$, Equation 9-8 approximates to Equation 9-9 where the largest eigenvalue λ_L of the adjacency matrix $\underline{A}$ and its corresponding eigenvector x_L are used to determine the vector of EVC values for the graph vertices (Dehmer, 2010).

$$\underline{\gamma} = \frac{1}{\lambda_L} A x_L \qquad \textit{Equation 9-9}$$

Hence, given a matching M, the $\underline{\gamma}$ can be determined. One can then use Equation 9-10 to determine the fairness of the IS network based on the value of χ. The value of χ is the number of important companies in the network, which is the size of the set N that contains the vertices whose EVC value lie within one standard deviation (σ_γ) of the mean EVC value $\overline{\gamma}$.

$$\chi = |N|, N = \{v_i, \qquad if\ \gamma_i - \overline{\gamma} \leq \sigma_\gamma \qquad \textit{Equation 9-10}$$

With this defined, it is possible to state the cost of fairness or a lack thereof. If a less fair network leads to long-term instability, then interventions must be made during the course of the IS to try and remedy any potential damage to partnerships. This may be in the form of meetings and re-drawing of contracts, all of which require negotiations. Furthermore, the lack of fairness and imbalance in company importance, companies can be exploited in a partnership. This exploitation is associated with transaction risk (Clemons et al., 1993).

With both the negotiations and transaction risk, transaction costs will increase since it can be split into negotiation costs (defined by Clemons et al. as coordination costs (1993)) and transaction risk (Clemons et al., 1993).

With this in mind, and the fact that smaller networks (in physical size and number of participants) reduce the transactional costs (Doménech and Davies, 2009), it is possible to determine the cost of fairness as follows:

$$Cost\ of\ fairness = TC \times (|m+n| - \chi) \times x_1 \times x_3 \qquad \textit{Equation 9-11}$$

TC is the standard transactional cost per unit distance for one partnership ($\frac{R}{km}$) and is increased by the number of companies that are significantly less important in the network ($|m+n| - \chi$), where $|m+n|$ is the number of companies in the IS. Additionally, the greater the number of connections and size of the network, the transactional costs will increase proportionally.

The cost of fairness ($\boldsymbol{f_4(x_1, x_3, x_4)}$) is the fourth cost used to calculate the index/cost of a distribution policy, as defined in Equation 9-2.

As mentioned previously, the cost of each distribution policy includes each of the matching feature costs ($f_j, j \in [1,4]$). However, the weighting the distribution policy gives to a matching feature cost f_j is ψ_j and therefore, a distribution policy can be uniquely profiled and quantified by the vector of weighting coefficients $\underline{\psi}$.

Table 9-3 summarises the vector of weighting coefficients for each distribution policy chosen for this study, while Table 9-4 lists the values for the TC and IC, the unit transactional costs and unit investment costs, respectively. It is important to note that in this chapter, the quantifying of the cost of a distribution policy is more about modelling approach and determining the suitability of an algorithm to a distribution policy than the accuracy in the determination of the actual costs. Specific values for the weighting coefficients and values for unit transactional costs (TC) and unit investment costs (IC) can be researched and accurately or empirically determined.

Table 9-3: Weighting coefficients for each distribution policy.

Distribution policy	ψ_1	ψ_2	ψ_3	ψ_4
Economically biased distribution policy (ECD)	4	20	1	1
Environmentally biased distribution policy (ECD)	2	500	1	1
Social construction-based distribution policy (SCD)	0.5	0.5	0.5	4

Table 9-4: Values chosen for the unit conversion cost factors, which would otherwise be empirically determined.

Unit conversion cost factor	Value
IC	0.15
TC	0.2

9.3 Graph representations for each algorithm

Now that the distribution policies have been quantified, the edge weights for the graphs for each algorithm can be defined. For the Simplex method, the weights of the graph edge e_{ij} is the k_{ij} in the objective function for the TP.

In general, the cost function for the weights can be defined as:

$$\begin{aligned} Edge\ weight = {} & Pricefactor_{ij} \times \sigma_{exchange} \\ & + RPD \times DistanceFactor_{ij} \times Payload\ Factor \end{aligned} \qquad \textit{Equation 9-12}$$

, where

$$Payload\ Factor = ceil\left(\frac{\min\left(abs\left(\sigma_{C_j}\right), \sigma_{S_i}\right)}{PAYLOAD}\right)$$

The $Payload\ Factor$ accounts for the number of trips a delivery truck must make since the amount of by-product material to be exchanged is limited by the payload of the truck. For this case study, a payload of 1 ton (1000kg) is chosen. The maximum amount possible to distribute between a supplier i and consumer j is $\min\left(abs\left(\sigma_{C_j}\right), \sigma_{S_i}\right)$. If this value is less than $PAYLOAD$, it will result in a $Payload\ Factor$ of 1. If the value is greater than the payload, then the nearest larger integer value will be chosen as the $Payload\ Factor$, which is taken care of by the function $ceil()$.

The only differences for the cost function for each algorithm are the terms $Pricefactor_{ij}$, $\sigma_{exchang}$ and $DistanceFactor_{ij}$, as shown in Table 9-5.

Table 9-5: Differences in the terms for the cost function for each algorithm.

Algorithm	Terms		
	$Pricefactor_{ij}$	$\sigma_{exchange}$	$DistanceFactor_{ij}$
HM	$abs\left(\alpha_{C_j} - \alpha_{S_i}\right)$	$abs\left(\sigma_{C_j} + \sigma_{S_i}\right)$	d_{ij}
FF	$MaxPrice - abs(\alpha_{S_i} - \alpha_{C_j})$	$\min\left(abs\left(\sigma_{C_j}\right), \sigma_{S_i}\right)$	$1.1 \times \max(\underline{D}) - d_{ij}$
SM	$MaxPrice - abs(\alpha_{S_i} - \alpha_{C_j})$	$Max(\sigma_S, abs(\sigma_C))$- $abs\left(\sigma_{C_j} + \sigma_{S_i}\right)$	$1.1 \times \max(\underline{D}) - d_{ij}$

With these terms defined for each algorithm, accordingly, IS networks solutions from each algorithm can be obtained. An order of the rows and column was chosen as follows:

$$
\begin{array}{cc}
 & \begin{array}{ccccccc} C_1 & C_2 & C_0 & C_6 & C_3 & C_5 & C_4 \end{array} \\
\underline{\acute{A}} = & \begin{bmatrix} b_{41} & b_{42} & \cdots & \cdots & \cdots & \cdots & b_{44} \\ \vdots & b_{02} & \cdots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & b_{26} & \cdots & \cdots & \vdots \\ b_{31} & \dots & \ddots & b_{36} & \ddots & \cdots & \vdots \\ \dots & \cdots & \cdots & \cdots & \cdots & \cdots & b_{14} \end{bmatrix} \begin{array}{c} S_4 \\ S_0 \\ S_2 \\ S_3 \\ S_1 \end{array}
\end{array}
$$

9.4 Results

9.4.1 Matching results

For the given row and column order of the assignment matrix $\underline{\acute{A}}$, an IS network for each algorithm was produced and is shown in Figure 9-3, Figure 9-4, and Figure 9-5. Alongside each network is the corresponding bipartite graph representation of the matching M. Confirming the rigorous study done in Chapter 7 and 8, Table 9-6 shows the key graph metric results for each algorithm, where the HM produces the lowest graph edges and the largest number of graph components.

Table 9-6: Graph metrics for the IS networks from each algorithm.

Algorithm	Num. Graph components	Num. Graph edges
HM	4	6
FF	1	9
SM	2	9

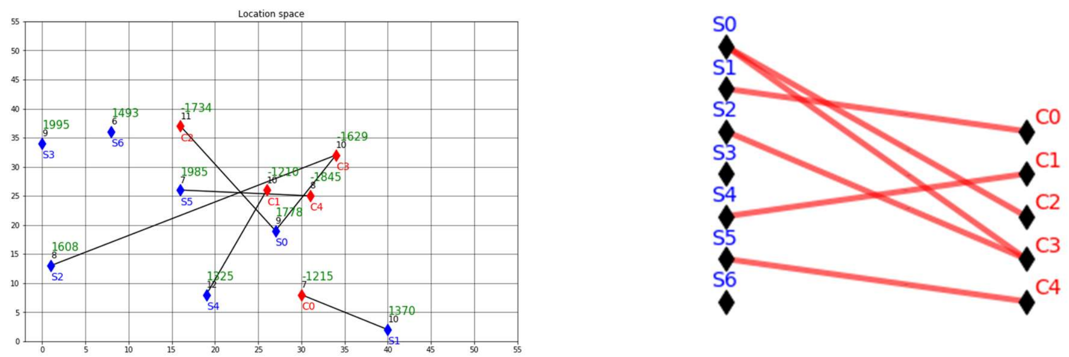


Figure 9-3:IS network (left) and matching M as a bipartite graph (right) from the HM algorithm.

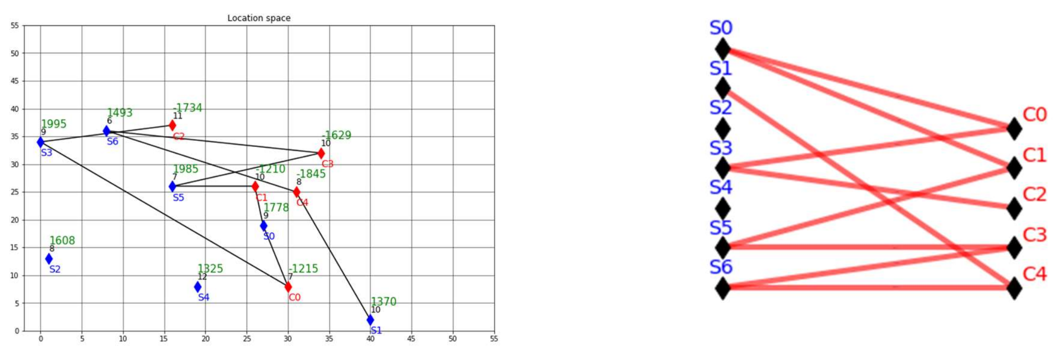


Figure 9-4: IS network (left) and the matching M as a bipartite graph (right) from FF algorithm.

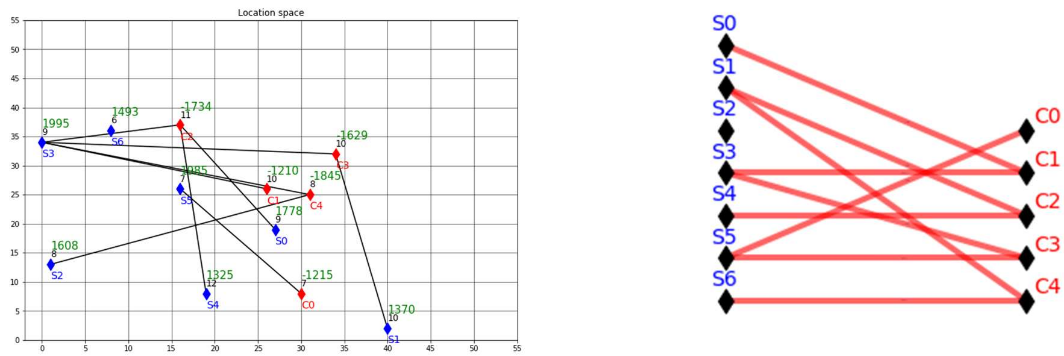


Figure 9-5: IS network (left) and the matching M as a bipartite graph (right) from SM method.

In this IS system, the total supply was 11554kg, and the total demand was 7633kg. This meant that at least one supplier would be unmatched and, therefore, not be part of the IS. The SM method omits 1 supplier (supplier 2) from the matching M whereas two suppliers are omitted from the HM IS network (supplier 3 and 6), and the FF IS network (supplier 2 and 4).

9.4.2 Matching feature results

Shown in Figure 9-6 are the EVC (γ) scatter plots for the HM (left), FF (right), and SM (centre). The red dotted lines bound a region that is within one standard deviation of the mean EVC value ($\overline{\gamma}$) for the IS network (shown as a blue horizontal line). The number of EVC data points that fall within this region corresponds to χ_i for the IS network of an algorithm i. The EVC values of a vertex is only shown in the scatter plot if it is not an isolated vertex (is within the Matching M). In other words, there is at least edge incident to the vertex. Any point in the scatter plot has a nonzero value, no matter how close it may be to the line $\gamma = 0$.

It is observed that χ_{HM}=10, $\chi_{FF} = 8$ and . $\chi_{SM} = 8$. Hence, the HM distributed by-product material in a way where a fairer IS network was produced. However, it was found that there is less dispersion in the γ values for the SM and FF IS network, as opposed to that in the HM IS network. When the RMSE was calculated for the γ datapoints from its mean γ, for the algorithm i, it was found that $RMSE_{\gamma^{SM}} = 0.125$, $RMSE_{\gamma^{FF}} = 0.102$ and $RMSE_{\gamma^{HM}} = 0.166$. The lower RMSE values infer that the SM and FF create networks that are fairer since the level of influence of all vertices (companies), indicated by the γ values, are closer in value. However, a higher mean EVC value is reported for the HM IS network ($\overline{\gamma^{HM}} = 0.516$) compared to that of the SM IS network ($\overline{\gamma^{SM}}$=0.342) and FF IS network ($\overline{\gamma^{FF}} = 0.247$).

This is due to the greater number of graph components in an HM IS network as opposed to an IS network from the SM and FF. The number of companies in each subgraph for the HM IS network is smaller than the number of companies in a subgraph for the SM and FF IS networks. As a result, the vertex influence in each HM IS network subgraph is higher than it would be if they belonged to a subgraph with a larger number of companies (like those subgraphs in the FF and SM IS network).

Furthermore, it is seen in Figure 9-6 that the scatter plot for the HM (left) has relatively more vertices of equal γ value. This corresponds to the 1-1 supplier and consumer pairings for the HM (seen in Figure 9-3), which make up a subgraph. Therefore, while the SM seems to create a slightly fairer IS network than the HM, it is the simplicity in the network connections of an HM IS network subgraph that affords higher average vertex importance than that of an SM and FF IS network. This would suggest that the subgraphs of an IS network by the HM are fairer than that by the SM or FF.

As a result, subgraphs are created by the HM where companies have a smaller difference in their influence in the subgraph, which can be regarded as more stable as power dynamics are less likely to be experienced if more companies have a similar influence in their subgraph.

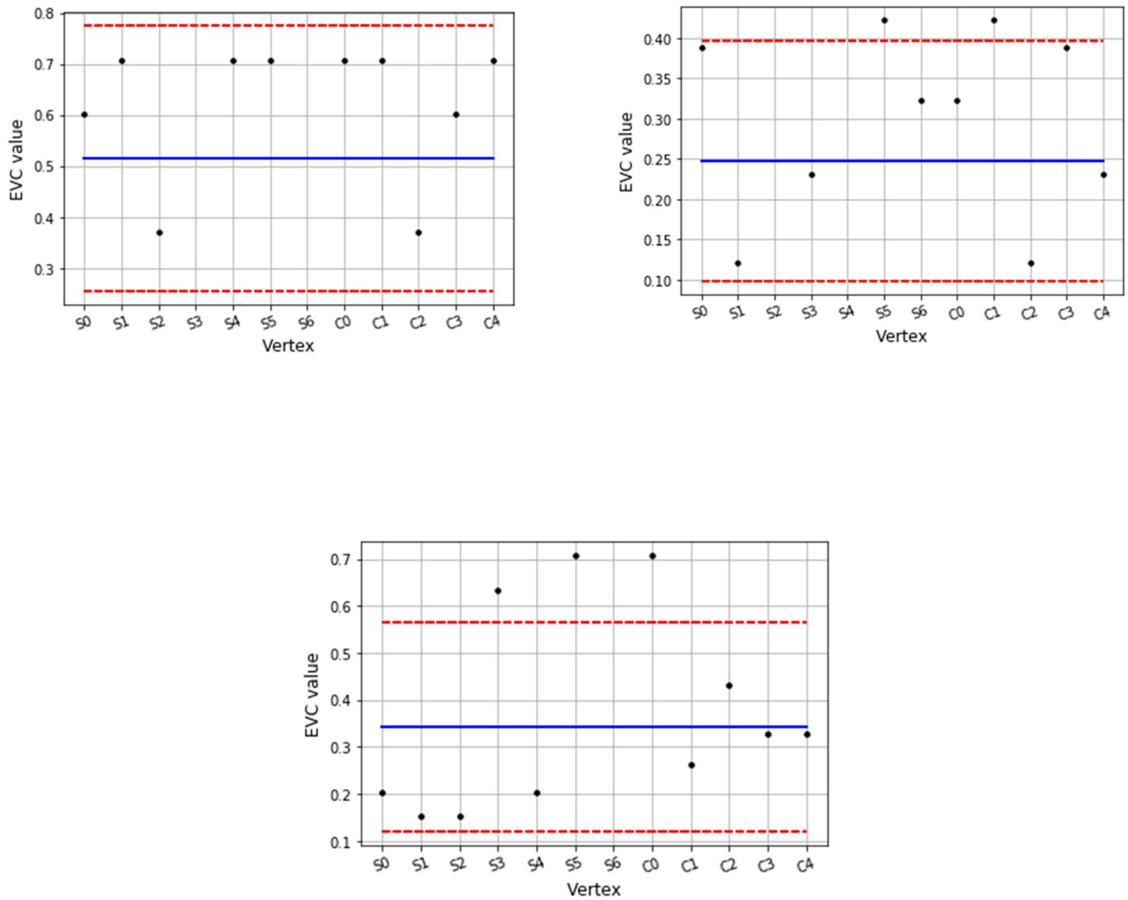


Figure 9-6: The χ results for the HM (top left), FF (top right) and SM (centre), show the number of vertices whose EVC value fall within 1 standard deviation of the mean EVC value.

The travelling distance for the HM-IS network was found to be the lowest compared to that of the other algorithms. The HM does produce a smaller IS network since the distance travelled in an HM IS network is 60.01% lower than that of an SM IS network and 49.12% lower than that of an FF IS network. This results in lower fuel consumption and CO_2 emissions, based on Equation 9-4. Table 9-7 summarises the matching feature results for each IS network generated by an algorithm.

Table 9-7: Matching feature results associated with the exchange of by-product material for each IS network produced by the algorithms.

Algorithm	x_1: Number of connections	x_2: Mass CO_2 emitted (kg)	x_3: Travelling distance (km)	x_4: Number of important companies
HM	6	2.99	313.41	10
FF	9	5.89	615.97	8
SM	9	7.49	783.73	8

The matching feature costs are shown in Table 9-8, and the overall cost of a distribution policy is shown in Table 9-9. It is found that in all distribution policies, the HM produces the lowest distribution index I_k for all distribution policies ($k = 1,2,3$).

Table 9-8: Results showing the matching feature costs from each algorithm.

	Matching feature cost (R)			
Algorithm	f_1	f_2	f_3	f_4
HM	1860.23	0.36	282.07	752.18
FF	3656.04	0.71	831.55	4434.96
SM	4651.79	0.90	1058.04	5642.86

Table 9-9: Cost of each distribution policy, in terms of its index, for each algorithm.

Distribution policy	Formula	Index (R)		
		HM	FF	SM
ECD	$4f_1+20f_2+f_3+f_4$	5158.21	8936.67	11370.65
EVD	$2f_1+500f_2+f_3+f_4$	4934.11	12931.74	16453.81
SCD	$0.5f_1+0.5f_2+0.5f_3+4f_4$	4080.06	19983.97	25426.79

This would imply that the HM is a better-suited algorithm in creating IS networks. It creates IS networks whose subnetworks (subgraphs) contain companies that have greater similarity in their network influence than if they belonged to those subgraphs from IS networks of SM and FF. Furthermore, with reflecting the same bias to the distribution policies in the cost function for the algorithms, the HM still manages to produce a more optimal IS network with respect to transportation costs (f_1), CO_2 emission tax (f_2), investment costs due to network complexity (f_3) and transactional costs (f_4).

By performing ten randomisations of the row and column order of the assignment matrix, similar to the approach used in the methodology of Chapter 7, one can determine how successful the HM is in being a better-suited algorithm. Ten randomisations of the row and column order were performed before the input matrix was processed by each algorithm. An iteration corresponds to a different randomisation of the row and column order before an algorithm processes the assignment matrix.

Table 9-10: Matching feature results for the HM.

Hungarian method				
Iteration	x_1	x_2	x_3	x_4
0	6	2.99	313.41	10
1	6	2.99	313.41	10
2	6	2.99	313.41	10
3	6	2.99	313.41	10
4	6	2.99	313.41	10
5	6	2.99	313.41	10
6	6	2.99	313.41	10
7	6	2.99	313.41	10
8	6	2.99	313.41	10
9	6	2.99	313.41	10

Table 9-11: Matching feature results for the FF algorithm.

Edmonds-Karp algorithm				
Iteration	x_1	x_2	x_3	x_4
0	9	5.96	624.17	8
1	9	6.74	704.99	8
2	9	4.77	498.76	8
3	10	8.14	851.61	8
4	9	7.12	745.16	6
5	9	4.26	445.83	9
6	9	7.06	738.46	8
7	9	5.70	596.34	8
8	9	3.69	386.11	9
9	9	5.89	615.97	8

Table 9-12: Matching feature results for the SM.

Simplex method				
Iteration	x_1	x_2	x_3	x_4
0	9	7.43	777.71	9
1	9	7.08	740.76	9
2	9	6.83	715.15	7
3	9	7.42	776.31	8
4	9	6.70	701.51	8
5	8	6.63	693.86	8
6	9	6.95	727.13	7
7	9	7.89	826.14	7
8	9	7.43	777.71	9
9	9	7.49	783.73	8

Table 9-10, Table 9-11 and Table 9-12 show the results for the matching features for the HM, FF, and SM, respectively. It is seen that in each iteration, the HM-IS network has the lowest number of connections (x_1), results in the lowest travelling distance (x_3) and lowest CO_2 emissions (x_3) than the IS networks for the other algorithms. Furthermore, there are more companies in an HM-IS Network whose EVC is within one standard deviation from the mean EVC value than those in an FF-IS network or SM-IS network. In fact, there seems to be no difference in the reported results for the HM. This is due to the fact that the HM produces the same matching M (same IS network structure) for each row and column order of the assignment matrix, whereas different IS networks result for the FF and SM. This was confirmed by determining the standard deviation in the travelling distance value among all iterations for each algorithm, which was found to be 0 km for the HM, 40.56 km for SM, and 138.17 km for FF.

Table 9-13: Mean graph metrics from each algorithm's IS network across the iterations.

Algorithm	Mean Graph metric results		
	Number of edges	Num. Graph comp.	Travelling distance (km)
HM	6	4	313.41
FF	9.1	0.9	620.74
SM	9	1	805.09

It is reasoned that because the system is small (7×5), there are only a few Pareto-optimal solutions with respect to the cost function. All the algorithms are common in that the final solution results in the deficit being the total supply and demand being minimised. However, the number of ways of distributing the material are numerous. Furthermore, the number of ways that result in the lowest cost function value (sum of the edge weights of matching M) is even smaller.

Table 9-14 shows the distribution index for all algorithm and distribution policy pairs, averaged across iterations. Hence, in terms of the distribution index, it is seen in Table 9-14 that, on average, for all distribution policies, the HM produces the lowest cost. The total sum of all the distribution costs from HM-IS networks is 27.5% and 25.1% of that from the FF-IS networks and SM-IS networks, respectively.

Table 9-14: Mean distribution index across all iterations for each algorithm and distribution policy pair.

Algorithm	Mean distribution index		
	I_{ECD}	I_{EVD}	I_{SCD}
HM	2901.66	4934.11	4080.06
FF	9205.35	13231.38	20891.81
SM	10839.41	15716.78	20891.81

As opposed to the HM, the SM and FF make a series of default supplier and consumer pairings, for reasons explained in Chapter 8. For the FF, since all paths from the super source to the super sink vertices in the flow network are of equal length, the order in which augmenting paths are chosen from a set of possible augmenting paths follows a default order. As such, the feasible FF IS networks produced will be of inferior optimality. This is evident in when one compares Table 9-10, Table 9-11 and Table 9-12, where the travelling distance and CO_2 emissions from SM-IS networks and FF-IS networks are greater at every iteration than those for the HM IS network.

For the SM-IS network results in Table 9-12, there is less variance in the IS networks compared to that of FF-IS networks in Table 9-11. This is due to the fact that while in Phase 1 of the SM the network pairings are made in a default manner, in Phase 2 it is not the case. The simplex algorithm finds the correct pairings in Phase 2, since the k_{ij} coefficients of the decision variables, also known as the edge weights of the graph, are introduced into the simplex tableau. However, the initial feasible solution (CPF solution) from which Phase 2 starts may not be the most optimal choice. Since the problem has many variables, the state space is multi-dimensional. Hence, the initial feasible solution may be on a hyperplane from which the final CPF solution visited is from a set of Pareto-optimal points with respect to the objective function value Z.

It is seen in Table 9-12 that an optimal CPF solution is reached, which produces a lower travelling distance (e.g., x_3 =693.86 km) in some cases, and in other cases, a higher travelling distance is rendered (e.g., x_3 =826 km). In each case, however, the objective function value Z for the TP linear program obtains the same value. It may be that the solution is more optimal with the $DistanceFactor_{ij}$ or the $PriceFactor_{ij}$, which would result in different connections. Regardless, it is clear that both the SM and FF algorithm do not consider creating IS networks with network graph structures similar to HM IS networks (greater number of graph components), as seen in Table 9-13. It may be Phase 1 of the SM and the default choice in augmenting paths for the FF algorithm that a set of Pareto-optimal solutions having graph metric values similar to the HM is not explored.

When one compares Table 9-10, Table 9-11 and Table 9-12, the HM comes out as the most suited algorithm for creating fair and stable IS networks since ten companies have γ values within 1 standard deviation of the mean EVC value $\overline{\gamma}$ whereas a lower χ value is reported for the FF and SM algorithm. The other 2 companies in the HM-IS network (supplier 3 and 6) have γ values of 0 as they are not in the matching M.

Table 9-15: Mean EVC vertex value for each iteration and algorithm pair.

Mean EVC value for each iteration and algorithm pair			
Iteration	HM	FF	SM
0	0.516	0.234	0.344
1	0.516	0.238	0.344
2	0.516	0.247	0.344
3	0.516	0.247	0.344
4	0.516	0.245	0.357
5	0.516	0.236	0.445
6	0.516	0.247	0.371
7	0.516	0.238	0.340
8	0.516	0.236	0.344
9	0.516	0.247	0.342

Table 9-16: RMSE for the EVC values for each iteration and algorithm pair.

Quantifying dispersion of EVC values using RMSE			
Iteration	HM	FF	SM
0	0.166	0.091	0.148
1	0.166	0.095	0.148
2	0.166	0.102	0.109
3	0.166	0.082	0.133
4	0.166	0.069	0.090
5	0.166	0.108	0.138
6	0.166	0.102	0.068
7	0.166	0.095	0.104
8	0.166	0.108	0.148
9	0.166	0.102	0.125

In all iterations, since the HM returns the same IS network for each column and row order of the assignment matrix, the same $RMSE_{\gamma^{HM}}$ and $\overline{\gamma^{HM}}$ results for each iteration, as shown in Table 9-16 and Table 9-15, respectively. These measurements are highest for the HM than the other algorithms. Hence, it can be concluded that while there is more dispersion in the EVC values for all the vertices of an HM IS network, its subgraphs contain vertices that have higher EVC values and are more similar than the vertices in the subgraphs for the FF and SM IS network. In other words, more stable subgraphs are created by the HM as opposed by the SM and FF.

This says much about IS networks in general. Specifically, that smaller IS networks containing a smaller number of companies, like the subgraphs in an HM IS network, are more likely to thrive since each company has more similarity in its network influence. These HM subnetworks are, therefore, simpler and less prone to excess network complexity. This is deemed particularly advantageous for the application of the HM in creating inter-company exchanges for water distribution networks, as it was noted by Nobel and Allen (2000) and Aviso et al. (2011) that excess network complexity is a problem for inter-company exchanges.

A study by Doménech et al.(2019) found that transportation costs and transactional costs are significant barriers for companies thinking of engaging in IS. Smaller networks, however, can help reduce the transactional costs since there are fewer actors/participants in the network (sub-network) (Doménech and Davies, 2009). This was confirmed by determining the distribution index for the SCD policy, where the transaction cost (f_4) was the lowest for the HM. In this regard, HM produces networks that are closer to optimality with respect to transportation costs, environmental impact, operability, and social aspects, such as trust, than the FF or SM. This is attributed to the bijection that the HM makes, which results in a lower number of graph edges and a greater number of graph components. With the determination of the distribution cost results and finding out that HM results in the lowest cost for all distribution policies, it is therefore concluded that HM is the most suitable algorithm applied in this study.

10 CONCLUSIONS

In this research, it was shown that the creation of optimal IS networks with respect to distribution policies is possible with fundamental and well-established graph-based algorithms. A graph-based modelling approach was used to model the IS problem in a way that captures multiple objectives such as economic, social and environmental objectives. This was accomplished by defining the companies as vertices and the possible connections between them as graph edges, with the strength of the connection being represented by the edge weight. The study was conducted to determine if structural features of an IS network can be used to evaluate the effect of distribution policies on IS networks created by graph matching algorithms when they are adapted to the IS problem. The IS network instance was the facilitated synergy identification across relatively large geographic distances, which contributes to Virtual Eco-industrial park studies.

Three network flow problems were successfully used to model the IS problem using a bipartite graph. These network flow problems are the transportation problem, optimal assignment problem, and maximal flow problem, each of which had an associated matching algorithm. The matching algorithms were adapted and used to solve a particular instance of IS, which was the creation of feasible by-product synergies between companies seeking to exchange a single waste by-product material.

The matching algorithms were the Hungarian method, Edmonds-Karp algorithm, and the 2-Phase simplex method, with the former two being graph-based methods and the last algorithm being a linear-algebraic method. The study successfully captured multiple objectives in the graph-theoretic representation, with constraints on the transportation costs, fairness in matchings and minimisation of waste by-product material being observed. All the fundamental algorithms were able to be adapted to solve the IS problem by taking as input an assignment matrix, which represents the bipartite graph of the problem. This afforded the approach simplicity as a solution strategy and allowed for the seamless extraction of structural network features. These network features, called graph metrics, brought to light graph characteristics that were seemingly unique to a matching algorithm.

Furthermore, it was found that multiple solutions existed due to the different IS networks produced. The graph metrics brought to light that for a given algorithm, the row and column order of the input assignment matrix affects the IS network produced. On the other hand, it was also found that for a given row and column order, the choice of algorithm used determines which IS network solution is rendered. It was postulated that this observation would be more prominent with larger IS networks and may provide more insight into the algorithm's suitability in IS network creation. To investigate this, a framework was devised.

The framework was produced to investigate the suitability of the algorithms in creating IS networks of different sizes. Of specific interest were the implications that the graph metrics had on the optimality of an IS network under different distribution policies. Three distribution policies were defined, each having a bias to different aspects of sustainability, i.e., the social impacts, environmental impacts, and economic impacts. This framework was applied before a rigorous study commenced, with several simulated IS scenarios being evaluated by each algorithm.

The results from the study found that the HM characteristically produces IS networks with a smaller travelling distance, having a smaller number of graph edges and a greater number of graph components, whereas the opposite was true for the FF algorithm. These graph characteristics were partly attributed to the bijection that the HM creates at each iteration in its application to the IS problem, where the rate at which pairings are made is lower than the other algorithms, which render more of a one-many mapping. This led to the HM being defined as a bijective bipartite matching algorithm. The HM results were positive in the light of IS as they had promising implications to transportation cost, reduced network complexity, and fairness in IS networks, all of which are imperative for the success of an IS network.

It was reasoned that the apparent superior ability of HM in creating feasible IS networks is due to the potent optimisation features of the SM and FF being rendered ineffective due to the graph representations. This was confirmed by determining the consistency in the graph metrics produced by each algorithm, which was found to be higher for the HM. Furthermore, the average time complexity for each algorithm was empirically determined, where it was found that the average time-complexity for the FF and SM were significantly closer to the worst-case time complexity than was the average time complexity to its worse time complexity for the HM. In other words, the SM and FF visit sub-optimal network states before arriving at a final IS network. This, coupled with the higher consistency in IS network solutions by HM proves the reasoning in the HM superiority as a matching algorithm.

A simulated study was then conducted where it provided the opportunity for the structural characteristics and their implications surrounding the feasibility of IS networks to be quantified under different distribution policies. An attempt at modelling distribution policies from matching features was made, with a distribution index indicating the optimality of an algorithm in creating IS networks.

It was found that under all defined distribution policies, the HM was the most performant, rendering IS networks that resulted in the lowest distribution index. The distribution index compactly confirmed that the HM-IS networks are the most optimal with respect to transportation costs, greenhouse gas emissions, and network complexity, compared to those networks from FF and SM. Furthermore, the fairness of HM-IS networks is greater and promotes trust compared to SM-IS networks and FF-IS networks. This was a promising result as it confirmed the results from the rigorous study and implied that graph metrics could be used as a criterion for evaluating networks and ranking algorithms. It can be conclusively said that the HM can be ranked the highest in its ability to create feasible IS networks.

11 RECOMMENDATIONS

It is acknowledged that there are several solution strategies that may be able to render more optimal IS networks with respect to multiple objectives. However, in most cases, Pareto-optimal solutions are found, which usually requires stakeholder input. This questions the need for complex solution strategies given that an optimal solution may not exist. While it was proven that feasible IS networks can be created with these graph matching algorithms used in this study, there are a few recommendations to progress this thematic graph-based solution strategy. For instance, it is possible to apply this method to IS scenarios involving multiple and different by-product waste materials.

Furthermore, future research can focus on pairing this solution method to empirical studies that provide values for important performance indicators, such as the unit investment cost per kilometre (IC) used in the case study. This would allow for the modelling of distribution policies to be done more thoroughly and may lead to a unifying theory in quantifying distribution policies in general.

One can also improve the performance of the FF algorithm in creating IS networks by altering the BFS algorithm to define shortest augmenting paths as those augmenting paths of the largest total edge weight. This would, in this particular instance, restore the greedy approach of the BFS algorithm, which was otherwise ineffective in a flow network with all augmenting paths having the same length. Furthermore, while the 2-phase simplex method was used to compare a linear-algebraic approach to graph-based approaches (such as HM), future work should determine if the HM is superior to the network simplex method, which is known to be more efficient than the 2-Phase simplex method.

Finally, it is acknowledged that IS network optimisation is not complete without considering time windows and route optimisation. While the aim of this study was to create feasible IS networks using a simpler approach, this study sets the foundation for future research to incorporate hybrid approaches that pair matching algorithms with route optimisation approaches. Specifically, it may be a more feasible situation if the optimisation of logistic operations is done in tandem with the optimisation of the IS network.

12 REFERENCES

Abdullah, N. & Hua, T. K. 2017. The application of the shortest path and maximum flow with bottleneck in traffic flow of Kota Kinabalu. *Journal of Computer Science & Computational Mathematics,* 7:(2)**,** 37-43.

Adhikari, B. 1960. Construction of socio-economic models for planning. *Eastern Anthropologist, India***,** 84-94.

Ahmed, F., Deng, J. & Tirkkonen, O. Self-organizing networks for 5G: Directional cell search in mmW networks. 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016. IEEE, 1-5.

Ahuja, R. K., Magnanti, T. L., Orlin, J. B. & Reddy, M. 1995. Applications of network optimization. *Handbooks in Operations Research and Management Science,* 7**,** 1-83.

Ahuja, R. K., Orlin, J. B., Sechi, G. M. & Zuddas, P. 1999. Algorithms for the simple equal flow problem. *Management Science,* 45:(10)**,** 1440-1455.

Al-Khamaiseh, K. & ALShagarin, S. 2014. A survey of string matching algorithms. *Int. J. Eng. Res. Appl,* 4:(7)**,** 144-156.

Alabi, O. A., Ologbonjaye, K., Awosolu, O. & Alalade, O. 2019. Public and environmental health effects of plastic wastes disposal: a review. *J Toxicol Risk Assess,* 5:(021)**,** 1-13.

Algergawy, A., Schallehn, E. & Saake, G. Understanding the schema matching problem. Proceedings of the 7th Conference on 7th WSEAS International Conference on Applied Computer Science, 2007. Citeseer, 59-68.

Amaral, L. A. & Ottino, J. M. 2004. Complex networks. *The European Physical Journal B,* 38:(2)**,** 147-162.

Andiappan, V., Tan, R. R. & Ng, D. K. 2016. An optimization-based negotiation framework for energy systems in an eco-industrial park. *Journal of Cleaner Production,* 129**,** 496-507.

Ardente, F., Cellura, M., Lo Brano, V. & Mistretta, M. 2010. Life cycle assessment-driven selection of industrial ecology strategies. *Integrated Environmental Assessment and Management: An International Journal,* 6:(1)**,** 52-60.

Athan, T. W. & Papalambros, P. Y. 1996. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization,* 27:(2)**,** 155-176.

Aviso, K. B., Chiu, A. S., Yu, K. D. S., Promentilla, M. A. B., Razon, L. F., Ubando, A. T., Sy, C. L. & Tan, R. R. 2015. P-graph for optimising industrial symbiotic networks. *Chemical Engineering Transactions.* Italian Association of Chemical Engineering-AIDIC**,** 1345-1350.

Aviso, K. B., Tan, R. R., Culaba, A. B., Foo, D. C. & Hallale, N. 2011. Fuzzy optimization of topologically constrained eco-industrial resource conservation networks with incomplete information. *Engineering Optimization,* 43:(3)**,** 257-279.

Azapagic, A. 1999. Life cycle assessment and its application to process selection, design and optimisation. *Chemical Engineering Journal,* 73:(1)**,** 1-21.

Baas, L. 2011. Planning and uncovering industrial symbiosis: comparing the Rotterdam and Östergötland regions. *Business Strategy and the Environment,* 20:(7)**,** 428-440.

Bagajewicz, M. & Rodera, H. 2000. Energy savings in the total site heat integration across many plants. *Computers & Chemical Engineering,* 24:(2-7)**,** 1237-1242.

Boix, M., Montastruc, L., Azzaro-Pantel, C. & Domenech, S. 2015. Optimization methods applied to the design of eco-industrial parks: a literature review. *Journal of Cleaner Production,* 87**,** 303-317.

Boix, M., Montastruc, L., Pibouleau, L., Azzaro-Pantel, C. & Domenech, S. 2012. Industrial water management by multiobjective optimization: from individual to collective solution through eco-industrial parks. *Journal of Cleaner Production,* 22:(1)**,** 85-97.

Boons, F. A. & Baas, L. W. 1997. Types of industrial ecology: the problem of coordination. *Journal of Cleaner Production,* 5:(1-2)**,** 79-86.

Borgatti, S. P. & Li, X. 2009. On social network analysis in a supply chain context. *Journal of Supply Chain Management,* 45:(2)**,** 5-22.

Brooke, A., Kendrick, D. & Meeraus, A. 1996. GAMS release 2.25; a user's guide. GAMS Development Corporation, Washington, DC (EUA).

Bunke, H. Graph matching: Theoretical foundations, algorithms, and applications. Proc. Vision Interface, 2000. 82-88.

BusinessTech. 2021. *Here is the official Petrol price for March* [Online]. BusinessTech,. Available: https://businesstech.co.za/news/energy/471864/here-is-the-official-petrol-price-for-march-2/ [Accessed 21 March 2021].

Cao, X., Wen, Z., Xu, J., De Clercq, D., Wang, Y. & Tao, Y. 2020. Many-objective optimization of technology implementation in the industrial symbiosis system based on a modified NSGA-III. *Journal of Cleaner Production,* 245**,** 118810.

Casparie, T. Beyond the New Normal. The European Petrochemical Association Virtual Annual Meeting 2020, 15 October 2020. Shell Chemicals.

Cefic; ECTA 2011. Guidelines for measuring and managing CO2 emission from freight transport operations. *Cefic Report.*

Chen, Z. & Wang, J. 2012. Heat, mass, and work exchange networks. *Frontiers of Chemical Science and Engineering,* 6:(4)**,** 484-502.

Chertow, M. & Ehrenfeld, J. 2012. Organizing self-organizing systems: Toward a theory of industrial symbiosis. *Journal of industrial ecology,* 16:(1)**,** 13-27.

Chertow, M. & Portlock, M. 2002. Developing industrial ecosystems: approaches, cases, and tools. *Yale School of Forestry & Environment Bulletin Series. 95.*

.

Chertow, M. R. 2007. "Uncovering" industrial symbiosis. *Journal of Industrial Ecology,* 11:(1)**,** 11-30.

Chew, I. M. L., Tan, R. R., Foo, D. C. Y. & Chiu, A. S. F. 2009. Game theory approach to the analysis of inter-plant water integration in an eco-industrial park. *Journal of Cleaner Production,* 17:(18)**,** 1611-1619.

Chopra, S. S. & Khanna, V. 2014. Understanding resilience in industrial symbiosis networks: Insights from network analysis. *Journal of Environmental Management,* 141**,** 86-94.

Christesen, I., Scott, J., Krrishnamohan, K., Gabric, A. & Heart, S. 1999. What is needed to encourage adoption of industrial ecology. *Australia (scaricabile da http://www. icnet. ic. gc. ca/base/bioeconomy/eco/concepts/key/% 7BChristesen% 7D-Needs-for-IE-Adoption. pdf)*.

Cimren, E., Fiksel, J., Posner, M. E. & Sikdar, K. 2011. Material flow optimization in by-product synergy networks. *Journal of Industrial Ecology,* 15:(2)**,** 315-332.

Clarck, J. 2003. Introduction SCI, London. . *In UK Workshop sponsored by Marks and Spencer and the Green Chemistry Network (Vol. 28)*.

Clemons, E. K., Reddi, S. P. & Row, M. C. 1993. The impact of information technology on the organization of economic activity: The "move to the middle" hypothesis. *Journal of Management Information Systems,* 10:(2)**,** 9-35.

Clift, R. & Druckman, A. 2015. *Taking Stock of Industrial Ecology*, Springer International Publishing.

Collins, R. J. & Ciesielski, S. K. 1994. *Recycling and use of waste materials and by-products in highway construction*.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. 2009. *Introduction to algorithms*, MIT press.

Cour, T., Srinivasan, P. & Shi, J. 2007. Balanced graph matching. *Advances in neural information processing systems,* 19**,** 313.

Dai, L., Zhou, N., Lv, Y., Cheng, Y., Wang, Y., Liu, Y., Cobb, K., Chen, P., Lei, H. & Ruan, R. 2021. Chemical upcycling of waste polyolefinic plastics to low-carbon synthetic naphtha for closing the plastic use loop. *Science of The Total Environment***,** 146897.

Dantzig, G. 2016. *Linear programming and extensions*, Princeton University Press.

Darnall, N., Welch, E. W. & Cho, S. K. 2019. Sustainable supply chains and regulatory policy. *Handbook on the Sustainable Supply Chain.* Edward Elgar Publishing.

De Carvalho, M. G., Laender, A. H., Gonçalves, M. A. & Da Silva, A. S. 2013. An evolutionary approach to complex schema matching. *Information Systems,* 38:(3)**,** 302-316.

Dehmer, M. 2010. *Structural analysis of complex networks*, Springer Science & Business Media.

Dhole, V. R. & Linnhoff, B. 1993. Total site targets for fuel, co-generation, emissions, and cooling. *Computers & Chemical Engineering,* 17**,** S101-S109.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische mathematik,* 1:(1)**,** 269-271.
Domenech, T., Bleischwitz, R., Doranova, A., Panayotopoulos, D. & Roman, L. 2019. Mapping Industrial Symbiosis Development in Europe_ typologies of networks, characteristics, performance and contribution to the Circular Economy. *Resources, conservation and recycling,* 141**,** 76-98.
Domenech, T. & Davies, M. 2011. Structure and morphology of industrial symbiosis networks: The case of Kalundborg. *Procedia-Social and Behavioral Sciences,* 10**,** 79-89.
Doménech, T. & Davies, M. 2009. The social aspects of industrial symbiosis: the application of social network analysis to industrial symbiosis networks. *Progress in Industrial Ecology, an International Journal,* 6:(1)**,** 68-99.
El-Halwagi, M. M. & Manousiouthakis, V. 1989. Synthesis of mass exchange networks. *AIChE Journal,* 35:(8)**,** 1233-1244.
Erol, P. & Thöming, J. 2005. ECO-design of reuse and recycling networks by multi-objective optimization. *Journal of Cleaner Production,* 13:(15)**,** 1492-1503.
Fàbregas Vázquez, H. 2015. Two-sided matching theory.
Federal Highway Administration. 2018. *Highway Statistics 2016* [Online]. Washington, DC: U.S. Department of Transportation. Available: https://www.fhwa.dot.gov/policyinformation/statistics/2016/vm1.cfm [Accessed 20 April 2021].
Fichtner, W., Frank, M. & Rentz, O. 2004. Inter-firm energy supply concepts: an option for cleaner energy production. *Journal of Cleaner Production,* 12:(8-10)**,** 891-899.
Finnveden, G., Hauschild, M. Z., Ekvall, T., Guinée, J., Heijungs, R., Hellweg, S., Koehler, A., Pennington, D. & Suh, S. 2009. Recent developments in life cycle assessment. *Journal of Environmental Management,* 91:(1)**,** 1-21.
Ford, L. R. & Fulkerson, D. R. 1956. Maximal flow through a network. *Canadian journal of Mathematics,* 8**,** 399-404.
Ford, L. R. & Fulkerson, D. R. 2009. Maximal flow through a network. *Classic papers in combinatorics.* Springer**,** 243-248.
Foulds, L. 1982. Graph Theory: A Survey of its Use in Operations Research. *New Zeal. Oper. Res,* 10**,** 35-65.
Fraccascia, L. & Giannoccaro, I. 2020. What, where, and how measuring industrial symbiosis: A reasoned taxonomy of relevant indicators. *Resources, conservation and recycling,* 157**,** 104799.
Friedrich, E. & Buckley, C. 2002. The use of life cycle assessment in the selection of water treatment processes. *Water Research Commission (WRC) Report,(1077/1),* 2.
Fröhlich, H., Wegner, J. K., Sieker, F. & Zell, A. Optimal assignment kernels for attributed molecular graphs. Proceedings of the 22nd international conference on Machine learning, 2005. 225-232.
Frosch, R. A. & Gallopoulos, N. E. 1989. Strategies for manufacturing. *Scientific American,* 261:(3)**,** 144-153.
Gale, D. & Shapley, L. S. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly,* 69:(1)**,** 9-15.
Gava, O., Bartolini, F., Venturi, F., Brunori, G., Zinnai, A. & Pardossi, A. 2018. A reflection of the use of the life cycle assessment tool for agri-food sustainability. *Towards Sustainable Global Food Systems***,** 211.
Gendreau, M. & Potvin, J.-Y. 2005. Metaheuristics in combinatorial optimization. *Annals of Operations Research,* 140:(1)**,** 189-213.
Gibbs, D. 2008. Industrial symbiosis and eco-industrial development: An introduction. *Geography Compass,* 2:(4)**,** 1138-1154.
Goldberg, A. V., Tardos, É. & Tarjan, R. 1989. Network flow algorithm. Cornell University Operations Research and Industrial Engineering.
Goldberger, J. & Tassa, T. 2008. A hierarchical clustering algorithm based on the Hungarian method. *Pattern Recognition Letters,* 29:(11)**,** 1632-1638.

Graedel, T. & Allenby, B. R. 2010. *Industrial Ecology and Sustainable Engineering: International Edition*, Pearson Education Inc., Upper Saddle River, Prentice Hall.

Grimes-Casey, H. G., Seager, T. P., Theis, T. L. & Powers, S. E. 2007. A game theory framework for cooperative management of refillable and disposable bottle lifecycles. *Journal of Cleaner Production,* 15:(17)**,** 1618-1627.

Hajebi, S., Roshani, E., Cardozo, N., Barrett, S., Clarke, A. & Clarke, S. 2015. Water distribution network sectorisation using graph theory and many-objective optimisation. *Journal of Hydroinformatics,* 18:(1)**,** 77-95.

Hajebi, S., Temate, S., Barrett, S., Clarke, A. & Clarke, S. 2014. Water distribution network sectorisation using structural graph partitioning and multi-objective optimization. *Procedia Engineering,* 89**,** 1144-1151.

Haraguchi, N., Martorano, B. & Sanfilippo, M. 2019. What factors drive successful industrialization? Evidence and implications for developing countries. *Structural Change and Economic Dynamics,* 49**,** 266-276.

Haslenda, H. & Jamaludin, M. 2011. Industry to industry by-products exchange network towards zero waste in palm oil refining processes. *Resources, Conservation and Recycling,* 55:(7)**,** 713-718.

Havenga, J. 2010. Logistics costs in South Africa–The case for macroeconomic measurement. *South African Journal of Economics,* 78:(4)**,** 460-476.

Hein, A. M., Jankovic, M., Farel, R., Sam, L. I. & Yannou, B. Modeling industrial symbiosis using design structure matrices. 17 th INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2015, 2015.

Heshmati, A. 2017. A Review of the Circular Economy and its Implementation. *International Journal of Green Economics,* 11:(3-4)**,** 251-288.

Hewes, A. K. & Lyons, D. I. 2008. The humanistic side of eco-industrial parks: champions and the role of trust. *Regional Studies,* 42:(10)**,** 1329-1342.

Hillier, F. S. & Lieberman, G. J. 2010. *Introduction to Operations Research*, McGraw-Hill Higher Education.

Hirata, K., Sakamoto, H., O'Young, L., Cheung, K.-Y. & Hui, C.-W. 2004. Multi-site utility integration—an industrial case study. *Computers & Chemical Engineering,* 28:(1-2)**,** 139-148.

Isafiade, A. J. & Short, M. 2019. Review of Mass Exchanger Network Synthesis Methodologies. *Chemical Engineering Transactions,* 76**,** 49-54.

Jato-Espino, D. & Ruiz-Puente, C. 2021. Bringing Facilitated Industrial Symbiosis and Game Theory together to strengthen waste exchange in industrial parks. *Science of The Total Environment***,** 145400.

Kadu, M. S., Gupta, R. & Bhave, P. R. 2008. Optimal design of water networks using a modified genetic algorithm with reduction in search space. *Journal of Water Resources Planning and Management,* 134:(2)**,** 147-160.

Kantor, I., Robineau, J.-L., Butun, H. & Marechal, F. 2020. A mixed-integer linear programming formulation for optimizing multi-scale material and energy integration. *Frontiers In Energy Research,* 8:(ARTICLE)**,** 49.

Karlsson, M. 2011. The MIND method: a decision support for optimization of industrial energy systems–principles and case studies. *Applied Energy,* 88:(3)**,** 577-589.

Karlsson, M. & Wolf, A. 2008. Using an optimization model to evaluate the economic benefits of industrial symbiosis in the forest industry. *Journal of Cleaner Production,* 16:(14)**,** 1536-1544.

Kastner, C. A., Lau, R. & Kraft, M. 2015. Quantitative tools for cultivating symbiosis in industrial parks; a literature review. *Applied Energy,* 155**,** 599-612.

Keckler, S. E. & Allen, D. T. 1998. Material reuse modeling: a case study of water reuse in an industrial park. *Journal of Industrial Ecology,* 2:(4)**,** 79-92.

Kessler, A., Ormsbee, L. & Shamir, U. 1990. A methodology for least-cost design of invulnerable water distribution networks. *Civil Engineering Systems,* 7:(1)**,** 20-28.

Khakzad, N., Landucci, G. & Reniers, G. 2017. Application of Graph Theory to Cost-Effective Fire Protection of Chemical Plants During Domino Effects. *Risk analysis,* 37:(9)**,** 1652-1667.

Khakzad, N. & Reniers, G. 2015. Using graph theory to analyze the vulnerability of process plants in the context of cascading effects. *Reliability Engineering & System Safety,* 143**,** 63-73.

Kim, J., Peng, Y., Ivezik, N. & Shin, J. 2011. An optimization approach for semantic-based XML schema matching. *International Journal of Trade, Economics, and Finance,* 2:(1)**,** 78-86.

Kim, S. H., Yoon, S.-G., Chae, S. H. & Park, S. 2010. Economic and environmental optimization of a multi-site utility network for an industrial complex. *Journal of environmental management,* 91:(3)**,** 690-705.

Klee, V. & Minty, G. J. 1972. How good is the simplex algorithm. *Inequalities,* 3:(3)**,** 159-175.

Klemeš, J. J., Varbanov, P. S., Walmsley, T. G. & Jia, X. 2018. New directions in the implementation of Pinch Methodology (PM). *Renewable and Sustainable Energy Reviews,* 98**,** 439-468.

Kriege, N. M., Giscard, P.-L., Bause, F. & Wilson, R. C. Computing optimal assignments in linear time for approximate graph matching. 2019 IEEE International Conference on Data Mining (ICDM), 2019. IEEE, 349-358.

Kruja, E., Marks, J., Blair, A. & Waters, R. A short note on the history of graph drawing. International Symposium on Graph Drawing, 2001. Springer, 272-286.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly,* 2:(1-2)**,** 83-97.

Kuhn, H. W. 1956. Variants of the Hungarian method for assignment problems. *Naval Research Logistics Quarterly,* 3:(4)**,** 253-258.

Kuhn, H. W. 2012. A tale of three eras: The discovery and rediscovery of the Hungarian Method. *European Journal of Operational Research,* 219:(3)**,** 641-651.

Kyi, M. T. & Naing, L. L. 2018. Application of Ford-Fulkerson Algorithm to Maximum Flow in Water Distribution Pipeline Network. *International Journal of Scientific and Research Publications,* 8:(12)**,** 306-310.

Lammich, P. & Sefidgar, S. R. Formalizing the edmonds-karp algorithm. International Conference on Interactive Theorem Proving, 2016. Springer, 219-234.

Lawal, M., Alwi, S. R. W., Manan, Z. A. & Ho, W. S. 2020. Industrial symbiosis tools—a review. *Journal of Cleaner Production***,** 124327.

Lee, D. 2012. Turning waste into by-product. *Manufacturing & Service Operations Management,* 14:(1)**,** 115-127.

Limleamthong, P. & Guillén-Gosálbez, G. 2017. Rigorous analysis of Pareto fronts in sustainability studies based on bilevel optimization: Application to the redesign of the UK electricity mix. *Journal of Cleaner Production,* 164**,** 1602-1613.

Linnhoff, B. & Flower, J. R. 1978. Synthesis of heat exchanger networks: I. Systematic generation of energy optimal networks. *AIChE Journal,* 24:(4)**,** 633-642.

Lovelady, E. M. & El-Halwagi, M. M. 2009. Design and integration of eco-industrial parks for managing water resources. *Environmental Progress & Sustainable Energy: An Official Publication of the American Institute of Chemical Engineers,* 28:(2)**,** 265-272.

Lütje, A. & Wohlgemuth, V. 2020. Requirements Engineering for an Industrial Symbiosis Tool for Industrial Parks Covering System Analysis, Transformation Simulation and Goal Setting. *Administrative Sciences,* 10:(1)**,** 10.

Madani, K. 2010. Game theory and water resources. *Journal of Hydrology,* 381:(3-4)**,** 225-238.

Martello, S. 2010. Jenő Egerváry: from the origins of the Hungarian algorithm to satellite communication. *Central European Journal of Operations Research,* 18:(1)**,** 47-58.

Martin, M. & Harris, S. 2018. Prospecting the sustainability implications of an emerging industrial symbiosis network. *Resources, Conservation and Recycling,* 138**,** 246-256.

Marton, S., Harvey, S. & Svensson, E. 2016. Investigating operability issues of heat integration for implementation in the oil refining industry. *Proceedings of eceee industrial efficiency*.

Mehlhorn, K. & Sanders, P. 2007. Algorithms and data structures. *The Basic Toolbox***,** 295.

Mizuno, R. & Ishida, Y. Constructing large-scale low-latency network from small optimal networks. 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), 2016. IEEE, 1-5.

Monge, A. E. 2000. Matching algorithms within a duplicate detection system. *IEEE Data Eng. Bull.,* 23:(4)**,** 14-20.

Mulrow, J. S., Derrible, S., Ashton, W. S. & Chopra, S. S. 2017. Industrial symbiosis at the facility scale. *Journal of Industrial Ecology,* 21:(3)**,** 559-571.

Nasir, S., Hussein, M. Z., Zainal, Z., Yusof, N. A., Zobir, S. A. M. & Alibe, I. M. 2019. Potential valorization of by-product materials from oil palm: A review of alternative and sustainable carbon sources for carbon-based nanomaterials synthesis. *BioResources,* 14:(1)**,** 2352-2388.

National Research Council 1999. *Industrial Environmental Performance Metrics: Challenges and Opportunities*, National Academies Press.

Nemati-Amirkolaii, K., Romdhana, H. & Lameloise, M.-L. 2019. Pinch methods for efficient use of water in food industry: A survey review. *Sustainability,* 11:(16)**,** 4492.

Neto, E. P. & Callou, G. An Approach Based on Ford-Fulkerson Algorithm to Optimize Network Bandwidth Usage. 2015 Brazilian Symposium on Computing Systems Engineering (SBESC), 2015. IEEE, 76-79.

Neves, A., Godina, R., Azevedo, S. G. & Matias, J. C. 2020. A comprehensive review of industrial symbiosis. *Journal of Cleaner Production,* 247**,** 119113.

Neves, A., Godina, R., G Azevedo, S., Pimentel, C. & CO Matias, J. 2019. The potential of industrial symbiosis: Case analysis and main drivers and barriers to its implementation. *Sustainability,* 11:(24)**,** 7095.

Nobel, C. & Allen, D. 2000. Using geographic information systems (GIS) in industrial water reuse modelling. *Process Safety and Environmental Protection,* 78:(4)**,** 295-303.

O'Rourke, D., Connelly, L. & Koshland, C. P. 1996. Industrial ecology: a critical review. *International Journal of Environment and Pollution,* 6:(2-3)**,** 89-112.

O’Carroll, S., Wallace, J., Pineo, C., Basson, L., Woodcock, J., Daniel, R. & Mouton, C. The Western Cape Industrial Symbiosis Programme (WISP): An innovative approach to resource efficiency and waste minimisation for South African businesses. Proceedings of the 20th WasteCon Conference, 2014. 6-10.

Olesen, S. & Polley, G. 1996. Dealing with plant geography and piping constraints in water network design. *Process safety and environmental protection,* 4:(74)**,** 273-276.

Palafox-Alcantar, P., Hunt, D. & Rogers, C. 2020. The complementary use of game theory for the circular economy: A review of waste management decision-making methods in civil engineering. *Waste Management,* 102**,** 598-612.

Parlar, M., Sharafali, M. & Goh, M. 2019. Optimal control and cooperative game theory based analysis of a by-product synergy system. *Journal of Cleaner Production,* 233**,** 731-742.

Plastics SA. 2019. *Key statistics from latest recycling report* [Online]. Plastics SA, . Available: https://www.plasticsinfo.co.za/2019/08/31/key-statistics-from-latest-recycling-report/#:~:text=Here%20are%20some%20of%20the,more%20than%20the%20previous%20year.&text=South%20Africa%20recycled%2046.3%25%20of,world%2Dleader%20in%20mechanical%20recycling. [Accessed 25 April 2021].

Raabe, B., Low, J. S. C., Juraschek, M., Herrmann, C., Tjandra, T. B., Ng, Y. T., Kurle, D., Cerdas, F., Lueckenga, J. & Yeo, Z. 2017. Collaboration platform for enabling industrial symbiosis: application of the by-product exchange network model. *Procedia Cirp,* 61**,** 263-268.

Rahimi, A. & García, J. M. 2017. Chemical recycling of waste plastics for new materials production. *Nature Reviews Chemistry,* 1:(6)**,** 1-11.

Ram, S., Tare, M. S., Aswath, P. B. & Ralegaonkar, R. V. 2020. Potential of Co-Fired Fly Ashes as a Construction Material–A Review.

Ramos, M. A., Rocafull, M., Boix, M., Aussel, D., Montastruc, L. & Domenech, S. 2018. Utility network optimization in eco-industrial parks by a multi-leader follower game methodology. *Computers & Chemical Engineering,* 112**,** 132-153.

Ren, J., Liang, H., Dong, L., Sun, L. & Gao, Z. 2016. Design for sustainability of industrial symbiosis based on emergy and multi-objective particle swarm optimization. *Science of the Total Environment,* 562**,** 789-801.

Richardson, D. & Thomson, R. 1996. Integrating thematic, geometric, and topological information in the generalization of road networks. *Cartographica: The International Journal for Geographic Information and Geovisualization,* 33:(1)**,** 75-83.

Rødseth, K. L. 2016. Environmental efficiency measurement and the materials balance condition reconsidered. *European Journal of Operational Research,* 250:(1)**,** 342-346.

Rokni, M. 2016. Introduction to Pinch Technology.
Roughgarden, T. 2016. CS261: A Second Course in Algorithms Lecture# 5: Minimum-Cost Bipartite Matching.
Rouvray, D. H. & Balaban, A. T. 1979. Chemical applications of graph theory. *Applications of Graph Theory,* 177**,** 155-156.
Schiller, F., Penn, A. S. & Basson, L. 2014. Analyzing networks in industrial ecology–a review of Social-Material Network Analyses. *Journal of Cleaner Production,* 76**,** 1-11.
Schrijver, A. 2005. On the history of combinatorial optimization (till 1960). *Handbooks in operations research and management science,* 12**,** 1-68.
Schwarz, E. J. & Steininger, K. W. 1997. Implementing nature's lesson: the industrial recycling network enhancing regional development. *Journal of Cleaner Production,* 5:(1-2)**,** 47-56.
Seay, J. R., Chen, W. T. & Ternes, M. E. 2020. Waste plastic: Challenges and opportunities for the chemical industry. *Chemical Engineering Progress,* 116:(11)**,** 22-29.
Shibin, K., Gunasekaran, A., Papadopoulos, T., Dubey, R. & Mishra, D. 2016. Sustainable consumption and production: need, challenges and further research directions. *International Journal of Process Management and Benchmarking,* 6:(4)**,** 447-468.
Singh, P. & Sharma, V. 2016. Integrated plastic waste management: environmental and improved health approaches. *Procedia Environmental Sciences,* 35**,** 692-700.
Skiena, S. 1990. Coloring Bipartite Graphs. *section § 5.5. 2 of Implementing Discrete Mathematics: Combinatorics and Graph Theory With Mathematica.* Addison-Wesley**,** 213.
Smith, J. C. & Taskin, Z. C. 2008. A tutorial guide to mixed-integer programming models and solution techniques. *Optimization in Medicine and Biology***,** 521-548.
Song, X., Geng, Y., Dong, H. & Chen, W. 2018. Social network analysis on industrial symbiosis: A case of Gujiao eco-industrial park. *Journal of Cleaner Production,* 193**,** 414-423.
Spekman, R. E., Kamauff, J. W. & Myhr, N. 1998. An empirical investigation into supply chain management: a perspective on partnerships. *Supply Chain Management: An International Journal*.
Stavropoulos, S., Wall, R. & Xu, Y. 2018. Environmental regulations and industrial competitiveness: evidence from China. *Applied Economics,* 50:(12)**,** 1378-1394.
Takama, N., Kuriyama, T., Shiroko, K. & Umeda, T. 1980. Optimal water allocation in a petroleum refinery. *Computers & Chemical Engineering,* 4:(4)**,** 251-258.
Tanimoto, S. L., Itai, A. & Rodeh, M. 1978. Some matching problems for bipartite graphs. *Journal of the ACM (JACM),* 25:(4)**,** 517-525.
Tardos, E. & Kleinberg, J. 2005. Algorithm Design. Pearson.
Tarjan, R. E. 1983. *Data structures and network algorithms*, SIAM.
Thie, P. R. & Keough, G. E. 2011. *An introduction to linear programming and game theory*, John Wiley & Sons.
Tisserant, A., Pauliuk, S., Merciai, S., Schmidt, J., Fry, J., Wood, R. & Tukker, A. 2017. Solid waste and the circular economy: a global analysis of waste treatment and waste footprints. *Journal of Industrial Ecology,* 21:(3)**,** 628-640.
Van Heerden, J., Blignaut, J., Bohlmann, H., Cartwright, A., Diederichs, N. & Mander, M. 2016. The economic and environmental effects of a carbon tax in South Africa: A dynamic CGE modelling approach. *South African Journal of Economic and Management Sciences,* 19:(5)**,** 714-732.
Vanderbei, R. J. 2020. Network flow problems. *International Series in Operations Research and Management Science.* Springer**,** 229-256.
Vergara, S. E. & Tchobanoglous, G. 2012. Municipal solid waste and the environment: a global perspective. *Annual Review of Environment and Resources,* 37**,** 277-309.
Verma, R., Vinoda, K., Papireddy, M. & Gowda, A. 2016. Toxic pollutants from plastic waste-a review. *Procedia Environmental Sciences,* 35**,** 701-708.
Walls, J. L. & Paquin, R. L. 2015. Organizational perspectives of industrial symbiosis: A review and synthesis. *Organization & Environment,* 28:(1)**,** 32-53.
Wang, Y. & Smith, R. 1994. Wastewater minimisation. *Chemical Engineering Science,* 49:(7)**,** 981-1006.
Wasserman, S. & Faust, K. 1994. Social network analysis: Methods and applications.

Weise, T., Zapf, M., Chiong, R. & Nebro, A. J. 2009. Why is optimization difficult? *Nature-inspired algorithms for optimisation.* Springer**,** 1-50.
Wilson, R. J. 2015. *Introduction to Graph Theory uPDF eBook*, Pearson Education.
Yazan, D. M., Yazdanpanah, V. & Fraccascia, L. 2020. Learning strategic cooperative behavior in industrial symbiosis: A game-theoretic approach integrated with agent-based simulation. *Business strategy and the environment,* 29:(5)**,** 2078-2091.
Yazdanpanah, V. & Yazan, D. M. 2018. Industrial symbiotic relations as cooperative games. *arXiv preprint arXiv:1802.01167*.
Yazdanpanah, V., Yazan, D. M. & Zijm, W. H. M. 2020. Coordinating multiagent industrial symbiosis. *arXiv preprint arXiv:2006.01784*.
Yeşilkaya, M., Daş, G. S. & Türker, A. K. 2020. A multi-objective multi-period mathematical model for an industrial symbiosis network based on the forest products industry. *Computers & Industrial Engineering,* 150**,** 106883.
Yogalakshmi, K. N. & Singh, S. 2020. Plastic Waste: Environmental Hazards, Its Biodegradation, and Challenges. *Bioremediation of Industrial Waste for Environmental Safety.* Springer**,** 99-133.
Zeng, Y., Wu, X. & Cao, J. 2014. Research and implementation of hungarian method based on the structure index reduction for dae systems. *Journal of Algorithms & Computational Technology,* 8:(2)**,** 219-231.
Zhang, Y., Zheng, H., Chen, B., Su, M. & Liu, G. 2015. A review of industrial symbiosis research: theory and methodology. *Frontiers of earth science,* 9:(1)**,** 91-104.
Zhang, Y., Zheng, H., Chen, B. & Yang, N. 2013. Social network analysis and network connectedness analysis for industrial symbiotic systems: model development and case study. *Frontiers of Earth Science,* 7:(2)**,** 169-181.
Zhang, Y., Zheng, H., Shi, H., Yu, X., Liu, G., Su, M., Li, Y. & Chai, Y. 2016. Network analysis of eight industrial symbiosis systems. *Frontiers of Earth Science,* 10:(2)**,** 352-365.
Zheng, F., Simpson, A. R., Zecchin, A. C. & Deuerlein, J. W. 2013. A graph decomposition-based approach for water distribution network optimization. *Water Resources Research,* 49:(4)**,** 2093-2109.

APPENDIX

APPENDIX A: Flowcharts for the Algorithms

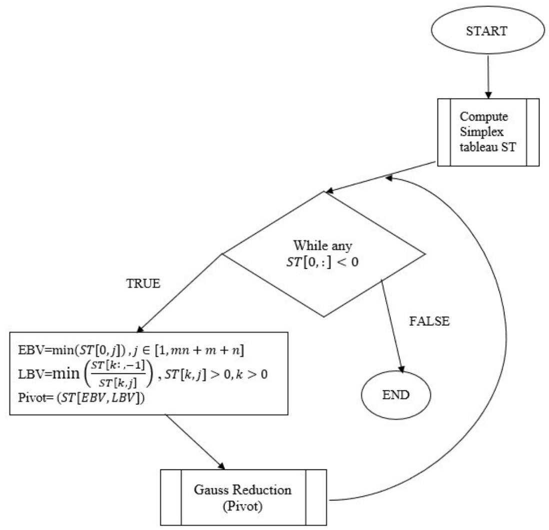


Figure 0-1: Flowchart of the Simplex algorithm.

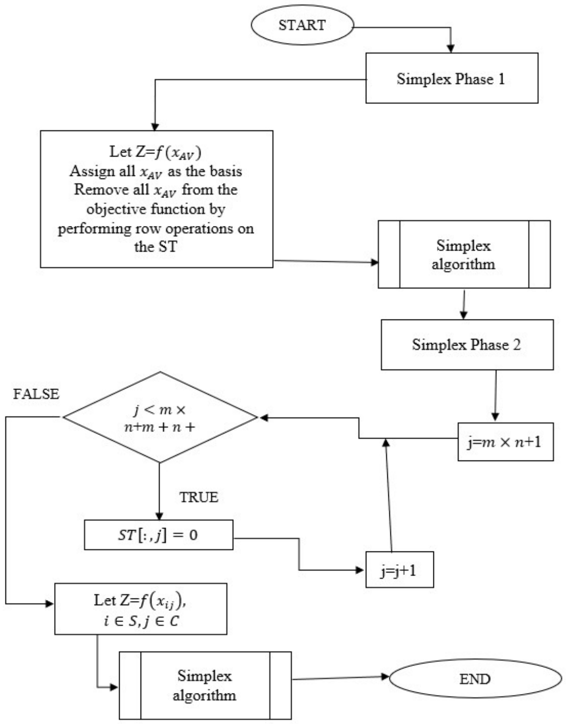


Figure 0-2: Flow chart of the 2-Phase Simplex method.

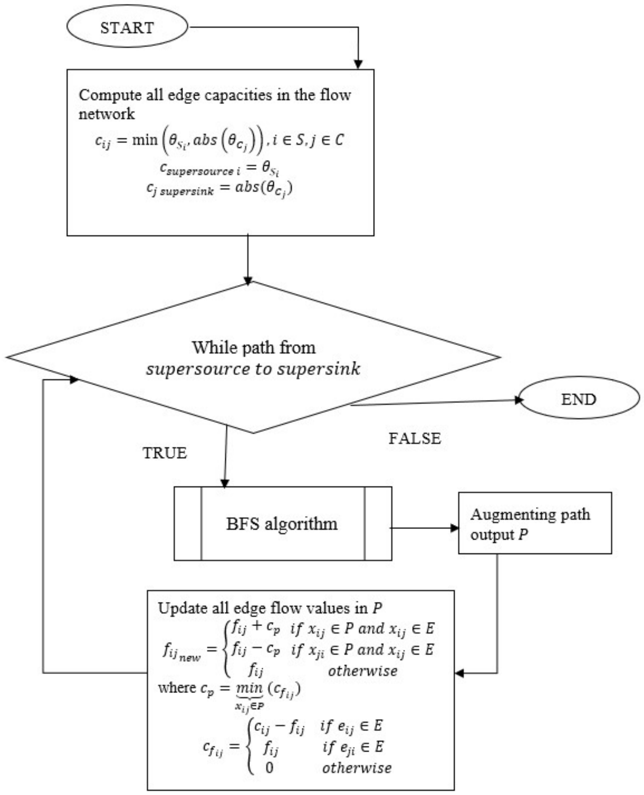


Figure 0-3: Flowchart for the Edmonds-Karp implementation in maximal flow problems.

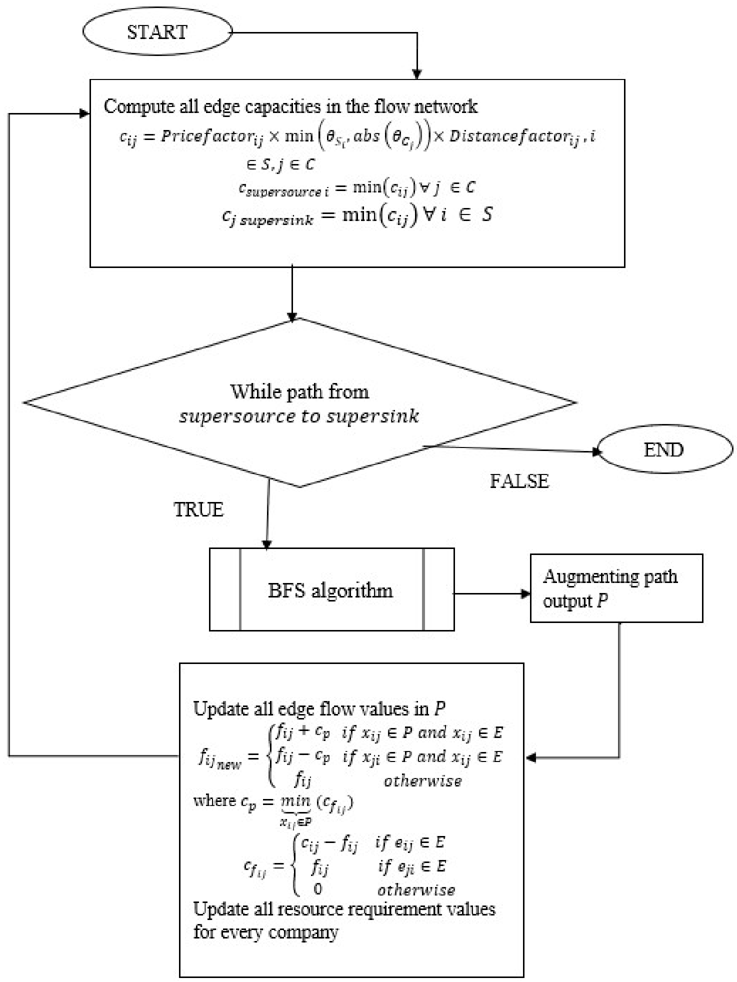


Figure 0-4: Flowchart of the Edmonds-Karp algorithm in IS for transportation cost minimization.

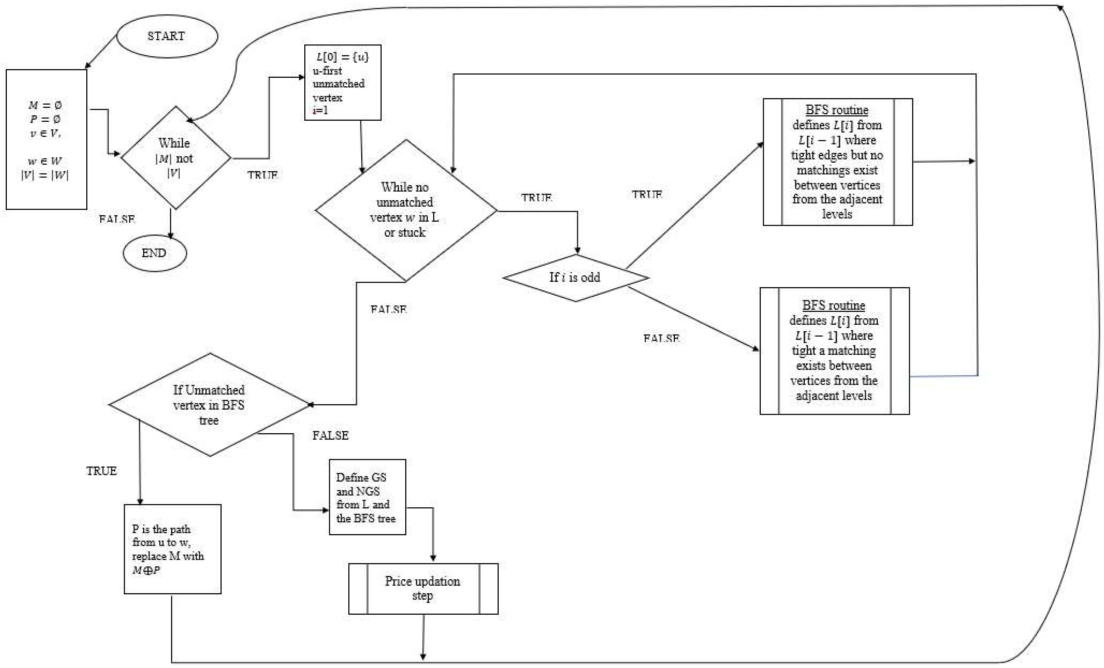


Figure 0-5:Flowchart of the Hungarian method.

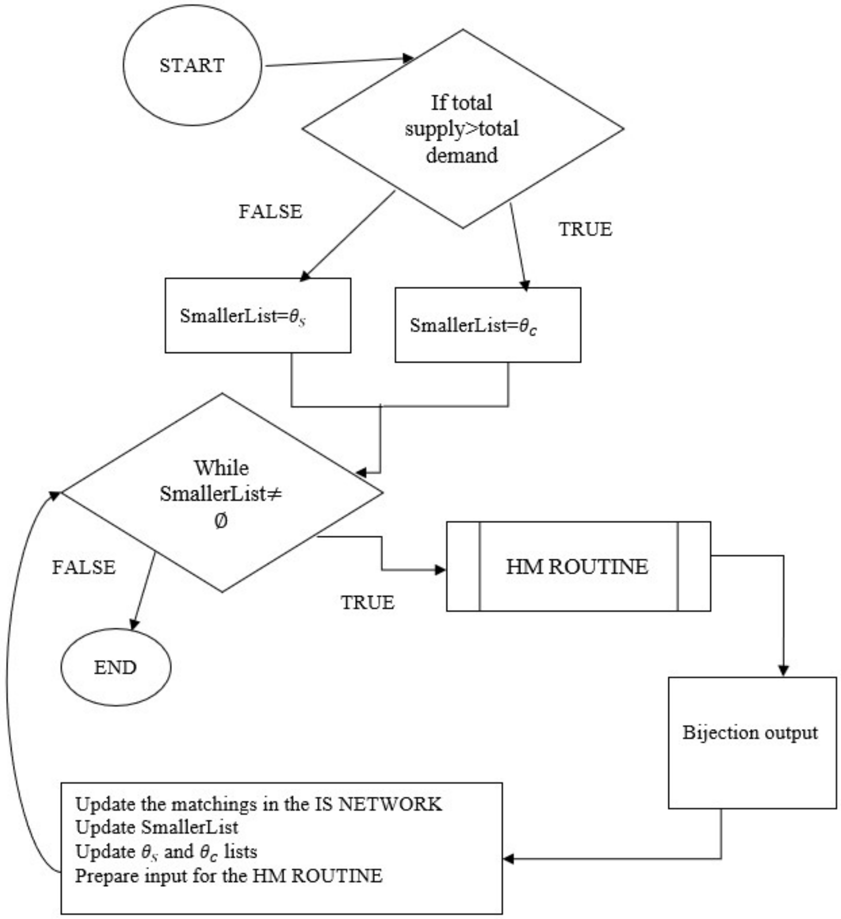


Figure 0-6: Flowchart for the Hungarian method implementation for IS.

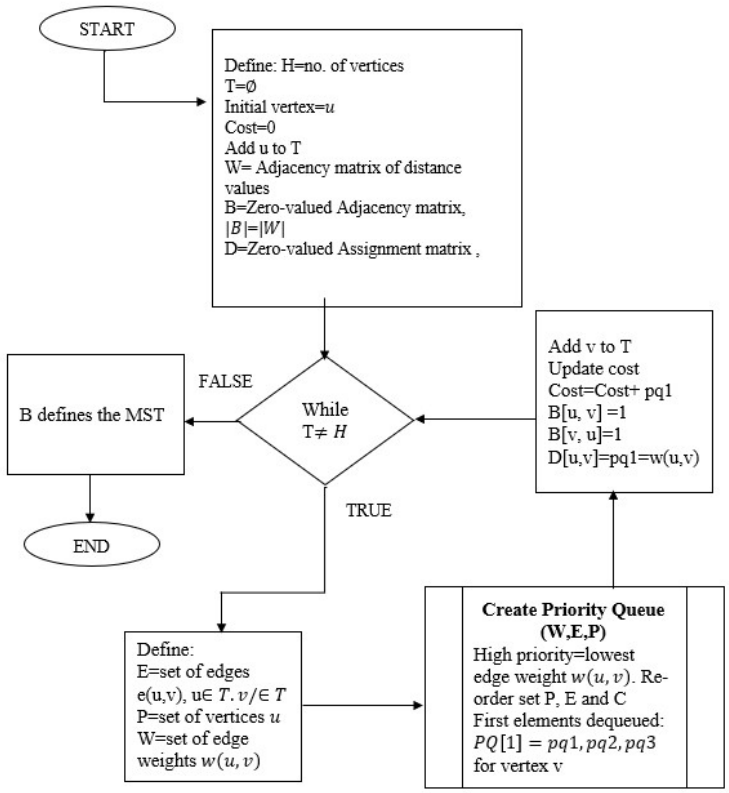


Figure 0-7: Flow Chart for Prim's Algorithm.

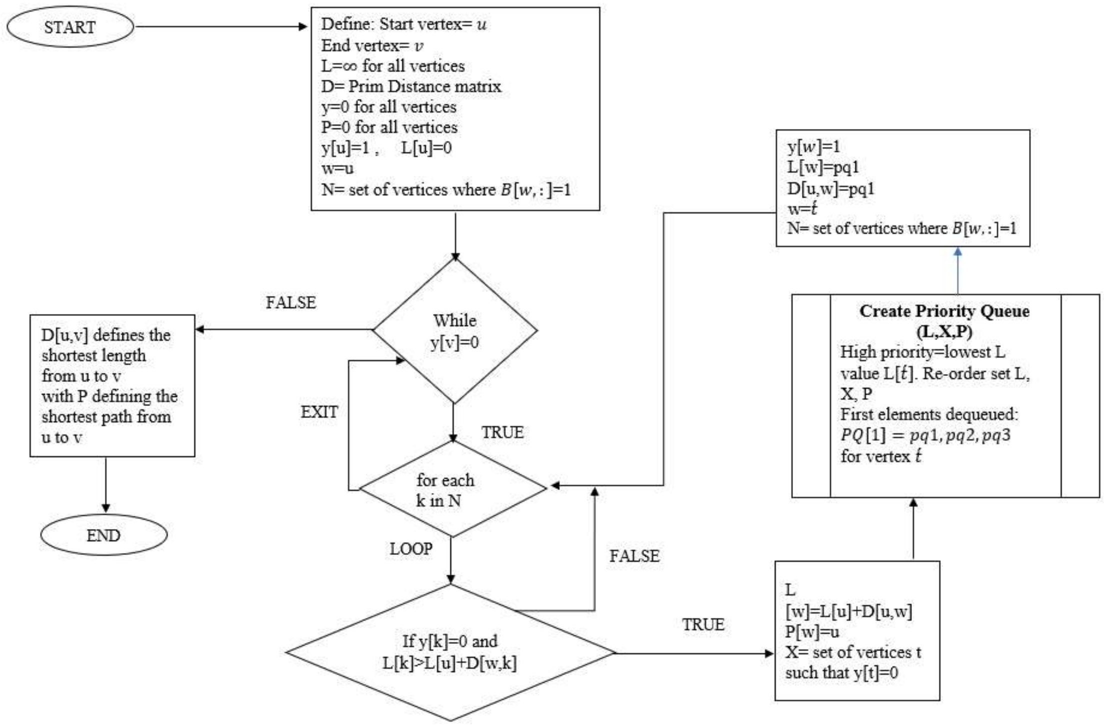


Figure 0-8: Flow chart for Dijkstra's algorithm.

APPENDIX B: NOMENCLATURE

Abbrev.	Description
BBMA	Bijective-Bipartite matching algorithm
BF	Basic Feasible Solution
BFS	Breadth-First Search
CP	Corner Point Solution
CPF	Corner Point Feasible Solution
DFS	Depth-First Search
EBV	Entering basic variable
ECD	Economically biased Distribution
EIP	Eco-Industrial Park
EVC	Eigenvector Centrality
EVD	Environmentally biased Distribution
FF	Edmonds-Karp algorithm
HEN	Heat Exchanger Networks
HM	Hungarian Method
IE	Industrial Ecology
IP	Industrial Park
IS	Industrial Symbiosis
LCA	Lifecycle Analysis
LP	Linear Programming
MEN	Mass Exchange Network
MILP	Mixed integer Linear Programming
MINLP	Mixed integer Non-Linear Programming
MP	Mathematical Programming
MSA	Mass Separating Agents
MST	Minimum Spanning Tree
MSW	Municipal Solid Waste
NLP	Non-Linear Programming
NPV	Net Present Value
OR	Operations Research
PA	Pinch Analysis
PM	Pinch Methodology
PMH	Population Metaheuristics
PP	Polypropylene
RMSE	Root Mean Square Error
SCD	Social Construction based Distribution
SLP	Standard Linear Program
SM	Simplex Method
SNA	Social Network Analysis
SSM	Single Solution Metaheuristics
ST	Simplex Table
STP	Standard Transportation Problem

TP	Transportation Problem
TSA	Total Site Analysis
VEIP	Virtual Eco-Industrial Park
WDN	Water Distribution Network
$\sigma, \alpha, \varphi, \theta$	Resource /Product attributes
d,C,τ	Constants defining the size of dataset groups
e_{ij}	Graph edge between vertex i and j
v_i	Vertex i