Plasma Chemistry and Plasma Processing (2023) 43:1013–1057
https://doi.org/10.1007/s11090-023-10351-w

ORIGINAL PAPER

Node-of-Influence Network Analysis for Targeted Condition Sequencing in Plasma Chemical Reaction Networks

Thomas D. Holmes[1] · Bryony C. Moody[2] · William B. J. Zimmerman[1]

Received: 14 January 2023 / Accepted: 30 May 2023 / Published online: 12 July 2023

Abstract

It was demonstrated that a plasma chemical reaction system can be represented as a directed bipartite variable-relationship (VR) graph to accurately represent node influence, and that targeted node-influence analysis of this graph can determine improved condition sequencing for a target outcome. A novel graph algorithm (OCARINA) was adapted for VR-graphs to give a measure of the net-influence of incremental increase (NIII) of one variable on another at any depth in the graph. Additionally, two conventional node-influence measures, the Eigenvector Centrality Index (ECI) and Katz Centrality Index (KCI), were also trialled on the VR-graph. The electron energy (ε) node influence was evaluated on a "baseline" continuous sequence of 10 ns 1 eV pulses in a 0D chemical-kinetic simulation using ECI, KCI and OCARINA NIII at three depths. KCI appeared to give meaningful values for ε influence in the whole graph but not on specific nodes, ECI gave no meaningful results. OCARINA (O_3 targeted) suggested each successive ε pulse had diminishing influence on O_3 formation, though analysis differed for each NIII depth. O_3 concentrations in simulations with different numbers of ε pulses decreased with each additional pulse, correlating with the OCARINA analysis. O_3, NO and O Species production in simulations of two ε pulses with one or both pulses incrementally changed by 10% from the baseline also largely agreed with the OCARINA results for each species on a baseline simulation of two consecutive 1 eV electron energy pulses. Additionally, it was found the NIII at antecedent depths corresponded to effects in subsequent phases in simulations.

✉ Thomas D. Holmes
t.d.holmes@sheffield.ac.uk

[1] Department of Chemical and Biological Engineering, The University of Sheffield, Sheffield, UK

[2] School of Mathematics and Statistics, The University of Sheffield, Sheffield, UK

Keywords Plasma chemistry · Graph theory · Optimization · Targeting · Chemical engineering

Introduction

Condition sequencing-The Staging of Input Conditions for a Desired Chemical Output

There is an ever-increasing variety of different plasma sources and power supply options for plasma chemistry, producing a range of different plasma discharges [1]. Though this results in a widening variety of different options, each option ultimately provides differing conditions (e.g. electric field strength, temperature, feed gas flow rate) in different sequences (e.g. 1000 °C then −50 °C then 25 °C etc.) to an Eularian spatial element of the plasma volume.

Which sequence of conditions is most appropriate depends on the required outcome (formation of a particular chemical, for example), and if an ideal condition sequencing for a given target outcome can be identified, this can be used as a basis for selecting the most suitable options to provide them.

For electron energy, for example, if the ideal sequence of electron energy can be known, it provides a focus for the required timing of voltage amplitudes, and therefore which power supply settings to use for the plasma source, thus narrowing down the vast array of potential options to a smaller number of more suitable contenders. In addition to electron energy (which tends to be the main controllable ‘input’ in non-thermal atmospheric-pressure plasma sources), there are also points when specific levels of chemicals, catalysts, heat, pressure, and almost any other variable in the system would be conducive to a desired outcome and could (if practical) also be sequentially altered accordingly.

This is also true for almost any chemical process. For a simple process or system, finding these points is essentially trivial. In chemical processing, the staging of operations in time or space–time is easy to represent and optimise for small reaction systems, the fed batch reactor or the intermediate inlets in a tubular reactor are well-known examples of designing by this principle [2] and for reactors-in-sequence, the Levenspiel plot [3]. But as processes/systems become increasingly complex the means of analysing them to find the ideal staging/sequencing of conditions need to be increasingly sophisticated. For networks with many reaction pathways and intermediate species, it is an open question. Presented here is an attempt to address it for plasma chemical reaction networks.

Increasingly complex systems mean increasingly greater numbers of entities which can influence each other in an increasing number of ways. Fortunately, it is possible for these influences between entities to effectively be mapped as a mathematical graph/network (the terms graph and network are used interchangeably in this work), and the resulting network topology to be analysed.

Mathematical graphs are composed of nodes and edges, with nodes being able to represent a variety of objects and edges representing the relationship between such objects. These graphs can hold a wealth of information, and additional information can be provided alongside the nodes and their relationships, to aid analysis. For example, it is possible to attribute direction [4] and weight to edges within a graph, as used in this research. Graph theory encompasses the study of these mathematical graphs and their properties, as well as algorithms that can be used to extract information from these graphs.

Graph theory has long been used in chemistry. A popular application is to model the synthesis of molecules [5, 6]. A further application is to model chemical reactions using graphs (it is within this area that our own research lies), and there are many approaches to doing this, including modelling exclusively the chemical species as nodes [7–10]. Other approaches separate the nodes into two distinct categories: species and reaction [11–14] and using edges to connect reactants to a specific reaction node for any given reaction, and then connecting the product nodes to this reaction as demonstrated in Holmes et al. [15], for application to plasma chemical reaction engineering. These species-reaction graphs fall into the category of a bipartite graph since they have two distinct sets of nodes (species and reaction) such that no node in one set is adjacent (connected by an edge) to any given node in the same set, only to nodes in the other set.

Reaction networks can also be modelled as hypergraphs [16]. These are graphs such that an edge does not only have to connect two nodes at a time. An edge might exit one node and partition into n edges which then connect to n new nodes. Fagerberg et al., [6] demonstrate that the directed graphs they use to model synthesis can be represented as hypergraphs. Current software packages, however, do not adequately support the visualisation and manipulation of hypergraphs for complex graphs (such as those we demonstrate in this work), though hypergraphs may be well worth considering for solving these types of problems in pending further developments in this area.

In plasma chemistry, graph theory has been used to filter reaction sets down to reduced-order plasma chemical reaction systems that can be used in the place of larger ones to significantly reduce computation time without losing any key functionality [17]. This is extremely useful for incorporating plasma chemistry into larger models (such as 2D or 3D time-dependent plasma simulations). Whilst not explicitly employing graph theory, there are other methods that deal with chemical reaction pathways based on the methods developed by Lehman [18] and adapted specifically to plasma chemistry with PumpKin [19]. These methods have been compared to graph theoretical alternatives [20] for identifying the principal pathways relating to a given species, and eliminating negligible pathways, reactions and species for computational expediency. Whether or not to apply such reduction methods depends on the purpose of modelling the chemistry. In chemical engineering, and engineering in general, one key use of modelling a system is to find the optimal conditions for a desired outcome. There is often a trade-off between having a model simple enough to do the number of parametric sweeps necessary to find these conditions, and having a model accurate enough that the results of such parametric sweeps will apply in reality. Even models with reduced reactions sets can still be complex enough to make a complete parametric sweep computationally unfeasible, and in the process, important information about potential key reactions may be lost, especially for reduction methods which eliminate reactions based on the roles they have been previously observed to play. Two chemical species, for example, may be extremely scarce, but when in the vicinity of each other may have a high probability of interaction (large collisional cross section) and form valuable products. Since this reaction would not contribute much to one or more simulations it might therefore be eliminated from future simulations for computational expediency. But since in many chemical engineering scenarios it is possible to inject reactants at any point in a process if necessary, or to catalyse reactions, if any such reactions have the potential to significantly influence a desired outcome, the obstacle of their low concentration of reactants may seem like an obstacle worth overcoming. Similarly, if any slow reactions form products that are potential key species for a desired outcome, means of increasing their rates may be worth looking into, even if other reactions tend to predominate in conventional conditions. In graph theory, a reaction (or species) such as this, if represented as a node, is a potential “node of influence” either on another node or number of nodes, or on the graph as a whole. Reactions and species with such potential can be elucidated by making plasma species-reaction graphs with edges weighted according to the reaction rate coefficients, as suggested in [15].

A node of influence is itself a flexible term. It could mean a node that directly influences another node, a node that influences the whole graph, a node that influences a subgraph of the graph, or a node that influences another node via its influence on the whole graph. The latter is of particular interest, but there do not seem to be many algorithms available that achieve this. This may partly be due to the types of graph used. Finding nodes of influence relies on having a graph where the influence of nodes on each other is accurately represented to begin with.

This work looks at the analysis for node-of-influence on target nodes and how this influence changes over the course of a 0D chemical kinetic simulation, with the aim of assessing how to alter these nodes to find improved condition sequencing for a target outcome. First, a more appropriate graph to represent node influence will be described in the next section. Following this, some key node-influence algorithms are discussed, including a modified version of the one in [15]. The next section describes the results of an investigation where a 0D simulation was run and the results of each time step were used to update a specific graph type on which, in turn, some node influence algorithms were run at regular intervals during the course of the simulation. Following these initial results, the results of the modified version of the algorithm in [15] were compared to the results of 0D simulations of different condition sequences (specifically of electron energy peaks) to assess the effect of altering nodes of influence according to the results of this node-influence algorithm.

Methods

Variable-Relationship graph–An alternative graph format to more accurately represent influence

Previously Species-Reaction graphs were used in Holmes et al. [15]. However, a species-reaction graph falls short of an accurate representation of all the nodes of influence for plasma chemistry, as arguably one of the greatest sources of influence in a plasma chemical system is the electron energy/electron temperature, which is neither a species or a reaction, and lies outside a species-reaction graph. The method described in the following section is an advance on this. In a simple mathematical description of chemical reactions (often used in reaction engineering), the species concentrations are variables, and the relationships between these variables are the reaction rate equations. Therefore, bipartite graphs of species and reactions can be generalised to graphs of variables and relationships. In the following example, for a reaction R which is

$$A + B \rightarrow C + D$$

the rate coefficient (k) is often a function of the electron temperature (T_e) in a plasma chemical reaction set, and the relationship of the change in concentration for C and D would be described by the following equation:

$$\Delta C \ or \ \Delta D = kAB\Delta t = f\left(T_e\right)AB\Delta t$$

And the relationship of the change in concentration for A and B would therefore be:

$$\Delta A \text{ or } \Delta B = kAB\Delta t = -f\left(T_e\right)AB\Delta t$$

A graph can be made from this relationship, with electron temperature also added as a variable node in the graph. This is shown in Fig. 1.

The node *R* in the centre of Fig. 1 is a *relationship node*, all other nodes in the figure are *variable nodes* and hence the graph is a *variable-relationship graph* (VR graph). Whether or not a variable is a dependent variable or an independent variable with respect to the relationship is represented by the direction of the edge (VR is an independent variable, and RV is a dependent variable). Due to the nature of equations expressing the concentration changes of reactants, it is necessary for the graph to be allowed negative edge weights as well as positive ones (e.g. edges RA and RB in Fig. 1).

The edge weights should represent the influence of the start node on the end node. For edge (A, R) or AR in Fig. 1, for example, the edge should constitute some form of description of the influence variable A has on relationship R. Clearly, the only effect of any influence on a scalar quantity is to increase or decrease its value, and the value of an equation will change depending on the value of its independent variables, and this value in-turn determines the value of the dependent variables, which are also independent variables of successive relationships, and so on.

A variable-relationship graph can be evaluated from the equations of its relationships to represent the *system in one or more states where all the variables have specific values*, or alternatively it could represent some measure of the *range of values (e.g. minimum to maximum) which each variable and relationship can take*. It can also be used without being evaluated simply to investigate the connections between the variables and relationships irrespective of their values (for simple measures of node connectivity, for example).

For graphs where the variables have specific values, edge weights can be set according to the partial differential of the start node with respect to the end node (Fig. 2). The result is a dynamic graph with the edge weights representing the instantaneous influence each node has on its successor. In the case of relationship-to-variable edges, the infinitesimal change in the dependent variable is identical to the infinitesimal change in the relationship, R, or identical to -R. Thus the ratio of the change in the value of the relationship to the change in the value of the dependent variable, and thus the edge weight,

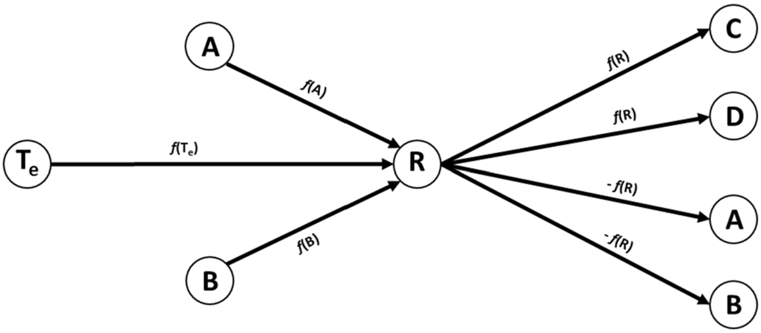

Fig. 1 A visual representation of a simple variable-relationship graph of the chemical equation A + B −> C + D as a directional graph **G** = ((A, R), (T_e, R), (B, R), (R, C), (R, D), (R, A), (R, B)) where R is the rate equation, and edge weights are functions of their start nodes with respect to their end nodes according to the equation R

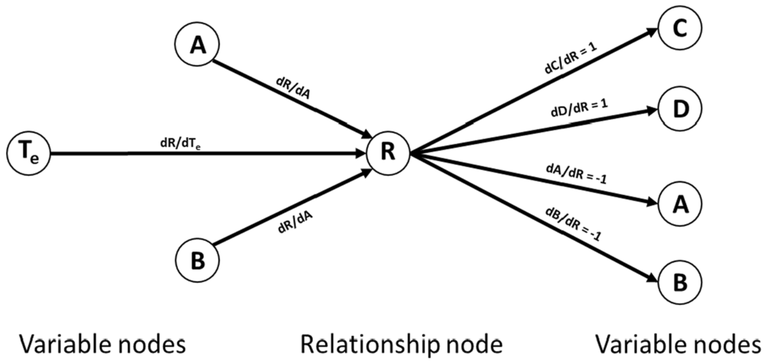

Fig. 2 A visual representation of a simple variable relationship graph of the chemical equation A + B – > C + D as a dynamic graph with the edge weights weighted as differentials

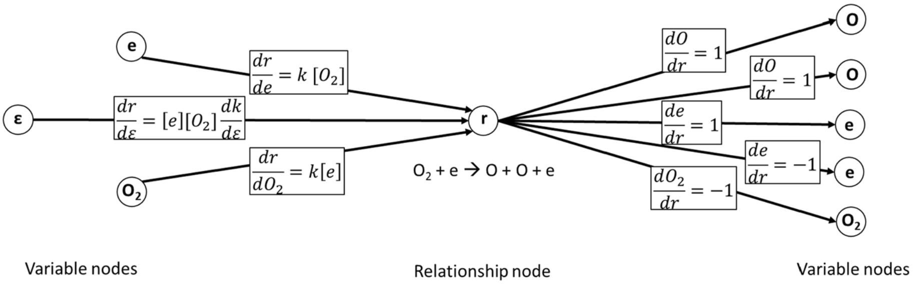

Fig. 3 A visual representation of a simple variable relationship graph of the chemical equation O_2 + e O + O + e as a dynamic graph with the edge weights weighted as differentials

should always be either 1 or -1. A more concrete example can be seen in Fig. 3 for the reaction O2 + e – > O + O + e, where: rate of reaction = r = k[e][O_2] and k = f(ε).

Another key piece of information is whether the influence of a variable on a relationship is always positive or always negative over a given value range, or always high or always low in relation to a defined value over a given range. Any of these four cases effectively means that although the value of the variable and relationship may change, this specific property (e.g. positive or negative, relatively high value or relatively low value, etc.) of the $V \rightarrow R$ edge in the system does not change for any possible scenario. Each of the plots in Fig. 4 gives some examples of range values. Such *range graphs*, can be used for algorithms where the system can be accurately represented over the given ranges. It may be necessary to use more than one number to accurately summarize the relationships in Fig. 4, (especially as the relationships are frequently functions of more than one variable), but various graph libraries (e.g. networkx) allow multiple edge attributes to be specified, so this should not necessarily be prohibitive.

It may also sometimes be useful to visualise these networks. In a network/graph visualisation, there are a number of scales that each of these values could be assigned to, e.g. *Edge thickness, Edge label, Edge label size, Colour Hue, Colour saturation, Colour brightness*. This can be done using software such as Gephi [21] as demonstrated in [15].

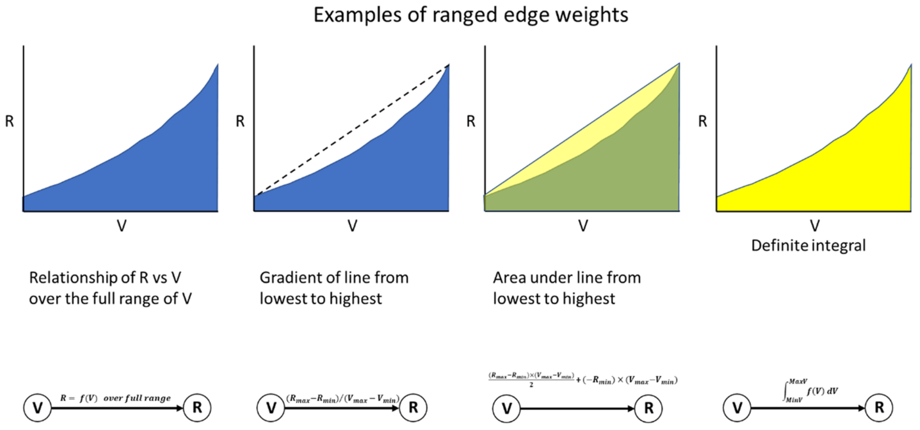
Examples of ranged edge weights
R
V
R
V
R
V
R
V
Definite integral
Relationship of R vs V over the full range of V
Gradient of line from lowest to highest
Area under line from lowest to highest
V
$R = f(V)$ over full range
R
V
$(R_{max} - R_{min})/(V_{max} - V_{min})$
R
$\frac{(R_{max} - R_{min}) \times (V_{max} - V_{min})}{2} + (-R_{min}) \times (V_{max} - V_{min})$
V
R
$\int_{MinV}^{MaxV} f(V)\, dV$
V
R

Fig. 4 Examples of different edge weights for the non-dynamic version of the graph of the equation A + B –> C + D. It could be thought of as a potential graph, as each of the edges are weighted according to the range between the maximum and minimum potential values that the variable can have

In addition to the tools available for visualisation, there is a wealth of algorithms available that could be used to analyse directed graphs and most programming languages have dedicated libraries for this purpose. Such algorithms are used to analyse mathematical graphs when the solution cannot be directly calculated, and as such, require heuristic methods or approximation algorithms. If such problems are hard to solve, but easy to verify a potential solution once it has been found, the problem is said to be NP-complete. The problem of maximising output in a chemical reaction network has been shown to be an NP-complete problem [22]. Regarding such problems, Fagerberg et al., [6] make the point that several good plans to choose from are better than one. Further to this, if all the options for improving a chemical synthesis can be presented (e.g. an algorithm could present them), then improvements to a current or baseline process can be made according to one’s judgement and the resources one has available, to a much greater extent than simply selecting a plan. In the context of this work, a plan can be thought of as an assembly of elements made of expected causes and effects (X causes Y, Y causes Z, etc.). If it is possible to consider all the elements of cause and effect then all the options of what could be conducive or destructive to a single objective could be laid out, allowing options to be selected that are best suited to available resources/capabilities. A graph can serve this purpose. The challenge is in identifying which elements are important to include in this “palette” of options, and which are not, and how to include them. Visualisation can be an important component in the ability to do this, but cannot be solely relied upon, especially when graphs become increasingly complex.

Graph analysis methods for nodes-of-influence

There are many existing graph algorithms which are all variations of centrality measures which rank nodes according to a measure of their importance within the graph. There are a variety of such measures, and they work to varying degrees of success depending on the graph they are being used on. Two of the most widely used of these methods are the Eigenvalue centrality index, and the Katz centrality index, which are both measures of node centrality that consider the importance of neighbour nodes as well as the node itself.

The importance of a node increases if it is itself connected to an important node [23]. This could be particularly convenient given the sequential nature of chemical reactions. For any given graph, a matrix A can be formed such that entry $a_{i,j}$ in the matrix is equal to one if there is an edge connecting node i and j. This matrix is called an adjacency matrix and can be further extended to weighted graphs by replacing the value of one with the weight of the edge connecting i and j. The adjacency matrix can be used in measures of centrality. In graph theory, measures of centrality allow us to determine how influential any given node is, relative to other nodes in the graph. Measures of centrality have been used for highlighting a node's importance in chemical reactions, such as Silva et al. [14], who use the out-degree centrality to highlight influential nodes. However, this measure does not take into account if a node is connected to other 'influential' nodes and so we argue that eigenvector centrality measures make for a more natural metric to use.

By taking the eigenvector that corresponds to the highest eigenvalue of the adjacency matrix, you obtain a weighted sum of not only the direct connections but the indirect connections of any given node [23].

For a graph $G := (V, E)$ with adjacency matrix $A = \left(a_{i,j}\right)$ the relative centrality score x_i of node i is given as:

$$x_i = \frac{1}{\lambda}\sum\nolimits_{j\in M(i)} x_j = \frac{1}{\lambda}\sum\nolimits_{j\in V} a_{i,j}x_j$$

such that $M(i)$ is the set of neighbours for node i and λ is constant. There are multiple variations of this measure. One such variation is the Katz Centrality index. In this case, the Katz centrality measure [24] is given by:

$$x_i = \alpha\sum\nolimits_{j\in M(i)} a_{i,j}x_j + \beta$$

such that α and β are positive constants and β contributes to all vertices and removes the possibility of a node being attributed null centrality if it is not strongly connected to other components of the graph.

Another widely used centrality method is the page rank algorithm, which is a subtle adjustment to the Katz centrality. This, along with the eigenvector and Katz centralities are all described in [25].

These centrality measures, however, are concerned with node influences on the whole network. If we are seeking to target a particular chemical species represented as a node, we are interested in node influences on a specified node via the whole network. Figure 5 outlines a variable-relationship graph of a simple hypothetical reaction network composed of 3 chemical reactions: R_1 = A B + C, R_2 = B + C E and A + B A + D, and how this graph would be operated on using this process. In the example in Fig. 5, ***D*** is the target node. How important each node is to the target node depends on how it positively or negatively influences the target node, either directly (if it is a predecessor) or indirectly (if it is a predecessor of a predecessor… etc.). Using Variable-Relationship graphs of the type specified in Fig. 2, this importance can be determined numerically.

This has been investigated to some degree in [15]. Ranged species-reaction graphs were also used in a graph algorithm aiming to make the first few steps towards a graph theory based waveform targeting algorithm based on the ranges of the rate coefficients over a given range of electron energies, which the authors named Optimal Condition Approaching via Reaction-In-Network Analysis (OCARINA). This was based on the following logic: "If a reaction creates the target it's desirable, if a reaction destroys a target it's undesirable." And by extension: "If a reaction creates something that creates the target it's desirable, and

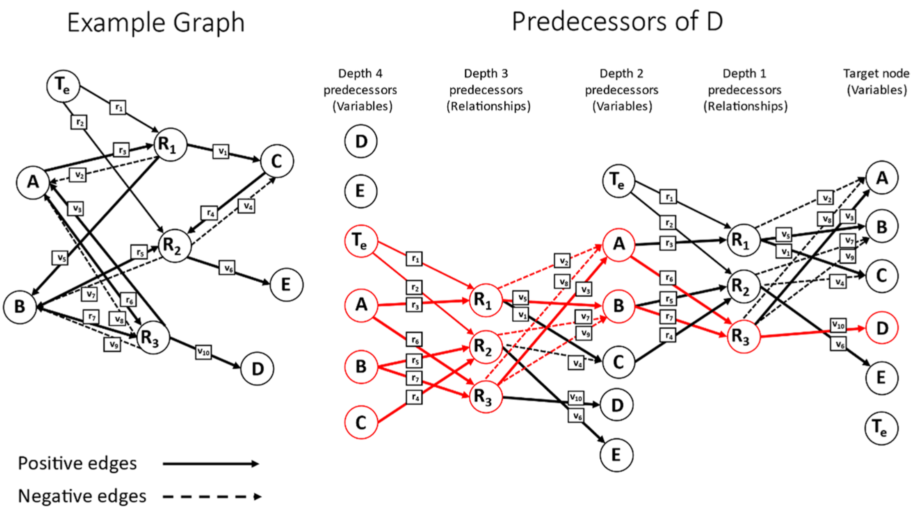
Example Graph
Predecessors of D
Depth 4 predecessors (Variables)
Depth 3 predecessors (Relationships)
Depth 2 predecessors (Variables)
Depth 1 predecessors (Relationships)
Target node (Variables)
Te
A
B
C
D
E
R1
R2
R3
r1
r2
r3
r4
r5
r6
r7
v1
v2
v3
v4
v5
v6
v7
v8
v9
v10
Positive edges
Negative edges

Fig. 5 (left) A diagram of a simple variable relationship graph composed of three chemical reactions. (right) A diagram of how a novel node of influence ranking algorithm would work on this variable relationship graph, the red lines showing the node influences on the target node D (Colour figure online)

if a reaction destroys something that creates the target, it's undesirable, etc.". The inclusion of negative edges (see Figs. 1, 2 and 3) in the VR graph, however, means this distinction of "negative and positive" is already made by the graph edges. Thus the influence of any node (the predecessor) on any other node (the target) can be measured by the sum of all the edges connecting the predecessor to the target, which can be performed on the adjacency matrix. The result will be a vector of the node influences (node ranking vector). The adjacency matrix can be multiplied by this vector, and summed over again for successive "depths" of predecessor any number of times (Fig. 5), which can be summarized as

$$x_i^{(n+1)} = \sum a_{i,j} x_i^{(n)}$$

where x_i is the *i*th column of the node ranking vector, $a_{i,j}$ is the *j*th row of the *i*th column, and n is the depth of predecessor from the target node. The initial node ranking vector (n = 0) with the index of each node set to 0, except the target node which is set to 1. "Depth" refers to the number of "steps back" (equal to the number of edges) between a predecessor node and the target node [26]. This method is explained in much more detail in the Appendix, including a validation of how the if–then statements in [15] determining the desirability of each reaction are exactly equivalent to this matrix operation.

Net influence of infinitesimal/incremental increase (NIII) – An accurate term for the OCARINA metric of node influence on dynamic VR graphs.

Here a more accurate term for the node-influence measure in [15] (when applied to a dynamic VR graph of the type shown in Figs. 2 and 3) is rationalised. The node-influence method in [15] returned the "demand" for a node based on the following logical propositions:

If a relationship ($\boldsymbol{R_i}$) positively affects $\boldsymbol{X}$, this relationship is desirable.
If a relationship ($\boldsymbol{R_i}$) negatively affects $\boldsymbol{X}$, this relationship is undesirable
etc.

However, the terms “desirability” or “demand” (a term used in [15], “electron energy demand”), though the latter is appropriate for some chemical processes (e.g. chemical oxygen demand), could be misleading when interpreting the output data of this node influence method on a VR graph. The algorithmic process is the sum of all the positive predecessor in-edges and all the negative predecessor in-edges of the target node. The data generated depends on how the edges are weighted, and in the case of the dynamic graph, where the derivatives are used to weight the edges, the result could be thought of as the sum of the positive and negative differentials, or “net differential”. However, “differential” is already such a commonly used term that “net differential” could also be misleading when used in this context. Since a differential is the rate of change of one number with respect to another, which as mentioned above amounts to the influence of the independent variable on the dependent one, the term “*net influence*” would be more convenient. Given that such an ‘*influence***’** of a differential, by nature, is *infinitesimally* small, and that the change in the independent variable of a differential is always positive, and is thus an *increase* (from its current value to a higher value), and the influence itself is the sum of many individual influences, i.e. a *net influence*, an appropriate term might be “*net influence of infinitesimal increase*” or “relative net influence of infinitesimal increase” as the numbers are on a relative scale. A term like this should be of more assistance when interpreting results.

Clearly, making infinitesimal changes is not much use without an infinite amount of iterations, therefore the values of the net influences of infinitesimal increase that the algorithm produces, are best used to inform incremental changes as long as the following assumption is valid:

$$\textit{Net potential influence of incremental increase} \approx \textit{Net influence of infinitesimal increase}$$

In most cases this will clearly become less accurate the larger the difference is between the incremental increase and the infinitesimal increase. Thus, the smaller the increase the more likely it is to have the “desired” proportional effect on the target (as the gradient is less likely to change much over a short difference in value). In the same way, the gradient is also less likely to change much over short distances in the opposite direction, suggesting that decreasing the value of the variable if its influence is negative should have the same effect as increasing the value of the variable if its influence is positive. We can then amend the logical propositions in above to give actionable changes to be made to be made to the values of predecessors of the target node (Table 1) for future simulations, to increase the value of the target node. The following sections test this in an air plasma reaction set, on target nodes of O_3, O and NO, focussing on electron energy as the key independent variable of influence.

Testing the Application of VR Graph Node-Influence Analysis Methods (Eigenvector Centrality, Katz Centrality, and OCARINA) in Time-Dependent 0D Plasma Chemical Kinetic Simulations

The VR graph and node influence analysis methods described in the previous section were investigated by applying them to a 0D plasma chemical kinetic simulation to analyse the net influence of the electron energy node on the formation of target chemicals. Sakiyama et al., [27] published a list of rate coefficients for a DBD plasma in moist air, with rate coefficients as functions of electron energy and temperature, and these rate coefficients were used for the 0D simulations in this work. Electron energy peaks of the type approximating Dielectric Barrier Discharge (DBD) streamers [28] were simulated by a triangular peaks increasing and decreasing linearly to and from a peak value. The gas temperature was kept

Table 1 Interpretation criteria for Net influence of infinitesimal increase (NIII) results

Conditional statement	Category	Potential action
If the **NIII** of a variable ($\boldsymbol{V_i}$) on $\boldsymbol{X}$ is positive, incrementally increasing this variable is likely to increase **X**	Positive influence	Incremental increase
If the **NIII** of a variable ($\boldsymbol{V_i}$) on $\boldsymbol{X}$ is negative, incrementally increasing this relationship is likely to decrease **X**	Negative influence	Incremental decrease
It also follows that:		
The greater the **NIII** of a variable ($\boldsymbol{V_i}$) on $\boldsymbol{X}$, the greater the change in $\boldsymbol{X}$ will result from a change in $\boldsymbol{V_i}$, and therefore the stronger the influence		

at a constant 300 K to limit the focus of the study to the effects of varying the electron energy. The simulation was run for 3,200,000 timesteps, with each timestep being 1×10^{-13} seconds. Species concentrations and reaction rates were calculated for each time step and the results collected.

A dynamic variable-relationship graph (Fig. 2) was constructed by adding relationships and variables from a list of reactions in [27] in csv format to a directional graph object in python using the NetworkX python library [29]. The differentials of the relationships were taken using the SymPy python library [30] and evaluated from values of the corresponding variables and relationships at any specified time step in the simulation. A section of an ego network of the electron energy (labelled **ee**) with a depth of 2 is shown in Fig. 6 and part of the full graph is shown in Fig. 7 using graph visualisations generated using Gephi [21]. Reverse graphs were also generated where the directions of all the edges in the graph were reversed.

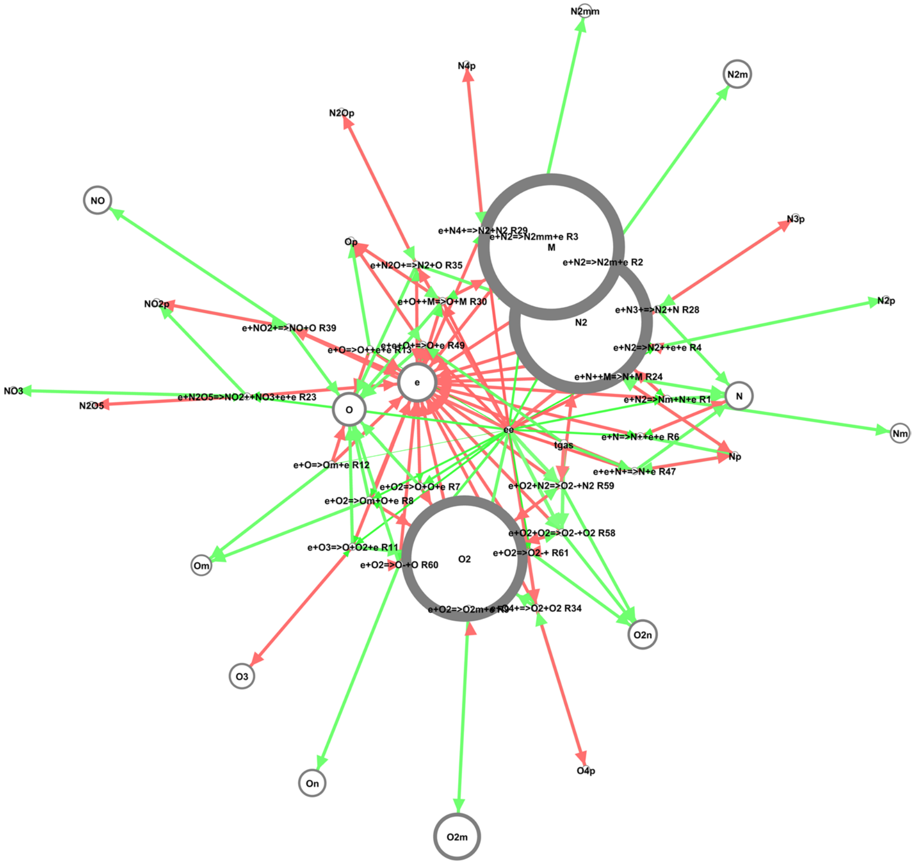

Fig. 6 Depth 2 ego network of electron energy (ee). Edges with positive values (positive influences) are coloured green and edges with negative values (negative influences) are coloured red. The edge thicknesses were also sized on a logarithmic scale according to their absolute values. The node diameters are sized logarithmically according to the variable value, Note that N2 and O2 size as proportional to their concentrations (Colour figure online)

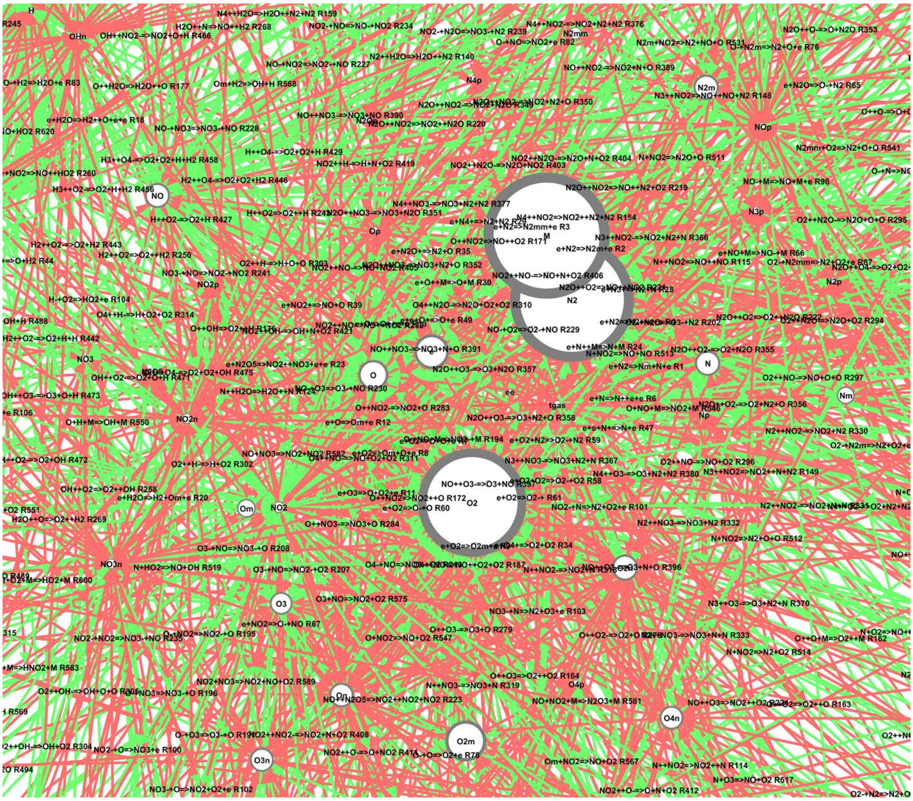

Fig. 7 The same ego network as in Fig. 6 but with the rest of the graph not hidden. The same size and colour criteria as Fig. 6 were used

For an initial trial of the method, the O_3 variable node was used as a target node since efficient and/or high yield ozone formation in plasma discharges has been extensively studied for over a century. Thus the ideal electron energy condition sequence for ozone formation has already been established and could be readily compared with the results of the net-influence analysis.

The Eigenvector Centrality Index (ECI), Katz Centrality Index (KCI) and OCARINA/Net-Influence of Infinitesimal Increase (NIII) were each generated for every 1000$^{\text{th}}$ time step in a "default condition sequence" of a continuous series of successive 1 eV electron energy peaks. The ECI and KCI were both generated from the graph using functions *eigenvector_centrality_numpy()* and *katz_centrality_numpy()* in the NetworkX library. Values of the graph edge weights needed to be scaled in order for the functions to work. A logarithmic scaling function was used:

$$n_{scaled} = 10^{\log_{10}(n/\alpha)}$$

where n is the number to be scaled, and α is the scaling factor (in this case 1000). If n < 0 then:

$$0 - n_{scaled} = 10^{\log_{10}(n/\alpha)}$$

was used. The OCARINA/NIII process did not require any scaling.

The results of this are shown in Figs. 9, 10 and 11, where any values are normalised, the normalisation function used was as follows:

$$n_{normalised} = \frac{n}{|n_{max}| + |n_{min}|}$$

The results of the NIII of electron energy on O_3 formation were then compared to different electron energy condition sequences to assess how the NIII compared to the condition sequences that gave rise to greater ozone formation. Two different sets of sequences were compared. A set where the number of successive 1 eV pulses was varied from 1 to 10 (Fig. 8a) and a set where the relative peak values of two successive pulses were increased or decreased by 0.1 eV increments (Fig. 8b).

Results

Node Influence Methods on a Default Condition Sequence of Continuous Electron Energy Peaks

The eigenvector centrality index did not appear to have any pattern (Fig. 9), this was likely due to the directional VR graphs containing nodes with no predecessors (in-degrees of 0). As such nodes will almost inevitably be present in VR graphs, the eigenvector centrality index is unlikely to be suitable for analysis of these graphs. The Katz centrality, however, solves this with the value beta, which is added to the centrality measures from the very start of the calculations, essentially giving each node a very small amount of importance to prevent nodes from being assigned a centrality of 0 even when they have high out-degree measures. For this reason there is a much clearer pattern in the Katz centrality results in Fig. 10. The Katz centrality of the electron energy node did not seem to vary much over the course of the simulation, however, the reverse Katz centrality (the Katz centrality on the

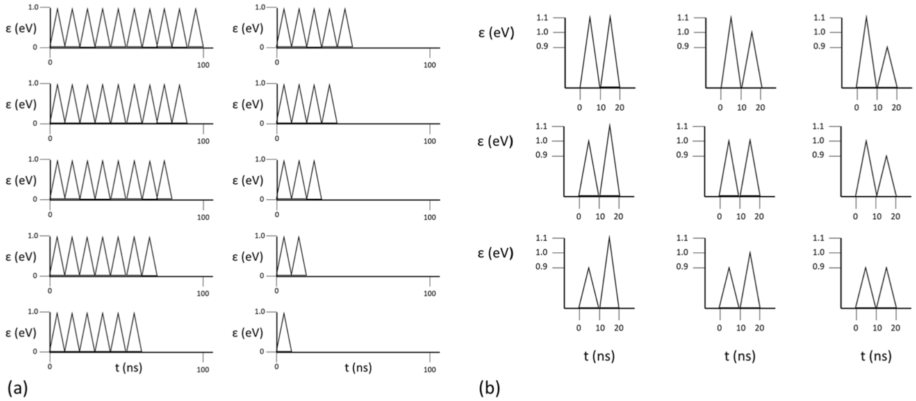

Fig. 8 **a** Different electron energy (ε) condition sequences 10 pulses of 0 or 1 eV. **b** Different electron energy (ε) condition sequences of two pulses with the peak values of one or both either increased or decreased by 0.1 eV

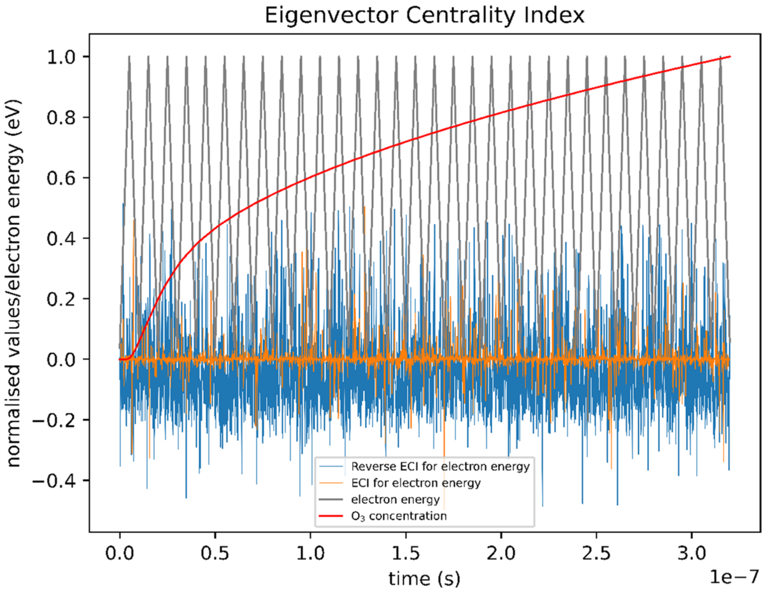

Fig. 9 A plot of the eigenvector centrality index and reverse eigenvector centrality index of a 0D plasma chemical kinetic simulation

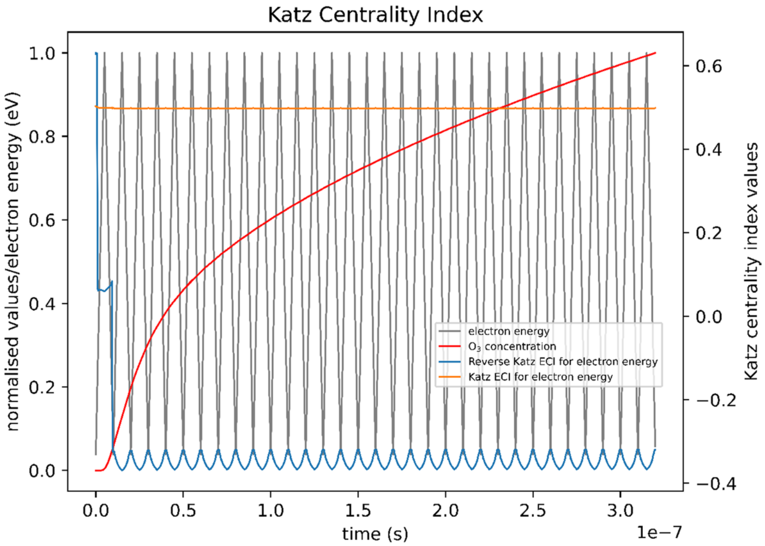

Fig. 10 A plot of the Katz centrality index and reverse Katz centrality index of a 0D plasma chemical kinetic simulation

reversed graph) of electron energy appeared to decrease sharply over the course of the first simulated electron energy pulse, and then again at the start of the second.

This could be explained by the rapid rates of electron impact reactions resulting from the increase in electron energy during the first peak, giving rise to increased reaction rates,

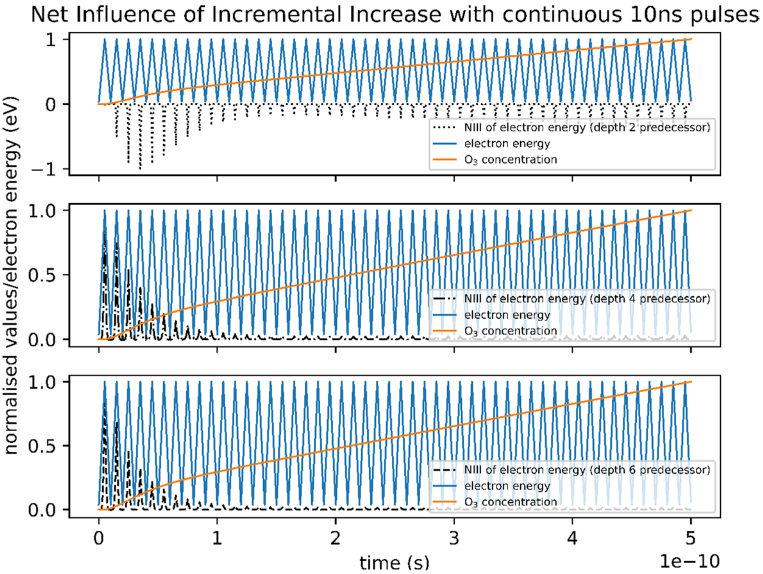

Fig. 11 Results of the OCARINA node-influence ranking algorithm of electron energy on influence on ozone formation over time in a 0D plasma chemical kinetic simulation of an air plasma, compared with the electron energy over time. The simulation simulated 10 ns triangular pulses of electron energy to simulate the effects of plasma streamers. Three depths of predecessor were analysed, direct precursors of O_3 (depth 2 predecessors) precursor of direct precursors of O_3 (depth 4 predecessors) and precursors of precursors of direct precursors of O_3 (depth 6 predecessors). Due to the differences in the orders of magnitude of the different precursor phases, the results were normalised to allow comparison of the trends

and hence edge weights. These initial electron impact reactions then produce products that undergo many further reactions that are independent of electron energy, hence the relative importance of the electron energy decreases. This suggests that the reverse Katz centrality index is representative of the influence of electron energy on the whole graph, and may be an appropriate method to use for assessing the change of node influence with time for VR graphs.

The NIII was evaluated at three preceding depths of variable nodes from a target node of O_3. These NIII values appeared in either positive or negative peaks which coincided with the peaks of electron energy (see Fig. 11). Interpreted using Table 1, this suggested the following:

For electron energy as a depth 2 (“direct”) predecessor to ozone, a positive change in the peak value of the first electron energy pulse would have a small net negative change in the value of the ozone concentration. After this first pulse, positive changes in the peak values of the electron energy pulses would have large net negative changes in the value of the ozone concentration.

For electron energy as a depth 4 predecessor to ozone, a positive change in the peak value of the first electron energy pulse would potentially have a relatively large net positive change in the value of the ozone concentration. Any positive change in the peak values of subsequent electron energy pulses would have increasingly less positive change in the value of ozone concentration, and this eventually levels off. Conversely, any equal and opposite changes to any of these pulses would have the equal and opposite effects.

I.e. decreasing the peak values of each successive electron energy pulse would have a less strongly negative effect. The trend in the NIII at depth 6, and its implications are the same as that of depth 4 in this case. Clearly, whilst depths 4 and 6 agree with each other, they are contradictory to depth 2, in that they suggest that increasing the electron energy is overall positive to O_3 formation, whereas depth 2 suggests the opposite. This discrepancy was fortunately resolved in the results interpretation of simulations with the condition sequences shown in Fig. 8b, which will be discussed in the next section.

There is also some agreement between the NIII at these three depths, however, in that the first electron energy peak is the least negative/most positive to O_3 formation. This is also reflected, at least to some extent, in the reverse Katz centrality which suggests that the initial electron energy peak has the most influence on the whole network, and successive peaks less so. In contrast to the sharp decreases of the Katz centrality of electron energy of the first two peaks, the NIII of electron energy on O_3 at depths 4 and 6 appear to show a more gradual decay in the positive influence of increasing the electron energy on O_3 formation. This suggests that each successive electron energy pulse may be giving diminishing returns for increasing the ozone concentration.

To test if this was indeed the case simulations were run with increasing numbers of electron energy pulses from 1 to 10 (see Fig. 8a), and each simulation was then left to run for a total of 3.2×10^{-7} s. The final O_3 concentrations of each simulation are plotted in Fig. 12, along with the final O_3 concentration per electron energy pulse as a measure of energy efficiency.

It can be seen in Fig. 12 that each additional electron energy pulse does indeed result in a progressively smaller increase in the ozone concentration in a trend strongly correlating with the first 10 "peaks" of NIII depths 4 and 6 of electron energy on O_3 in Fig. 11. The energy efficiency measure in Fig. 12 suggests a single pulse would be 336% as energy efficient as 10 successive identical pulses. This suggests that correct interpretation of the NIII results of a single "baseline" simulation, could anticipate a superior conditional sequence of pulses of 1 eV, 0 eV, 0 eV, 0 eV, 0 eV, 0 eV, 0 eV,… etc. for maximum output per input,

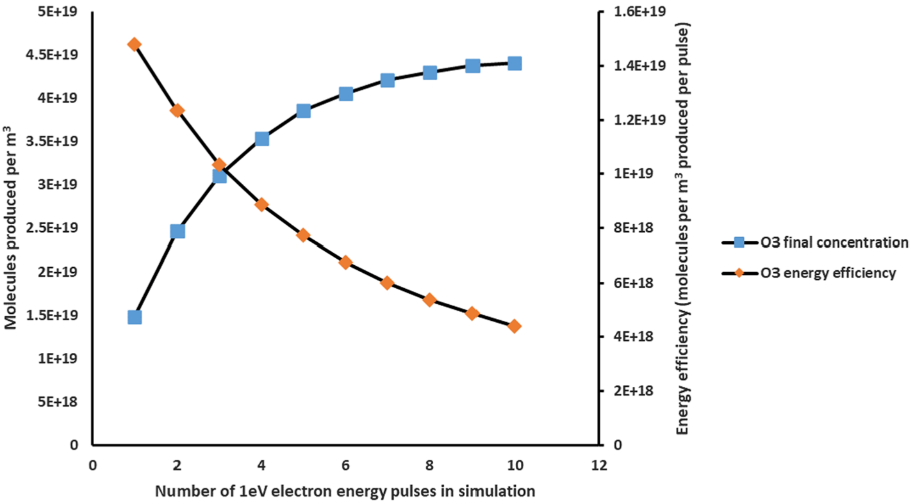

Fig. 12 Results of the final ozone concentrations and final ozone energy efficiencies of formation in ten different simulations of incrementally increasing numbers of streamers (from one to ten, see Fig. 8.a)

and a superior conditional sequence of 1 eV, 1 eV, 1 eV, 1 eV, 1 eV, 1 eV, 1 eV,… etc. for maximum output. The results also suggest that increasing the peak values of any of the pulses beyond 1 eV would also increase the O_3 formation, and decreasing it would have the opposite effect. However, Fig. 12 does not show the effects of all possible condition sequences of 10 pulses of peaks of either 1 eV and 0 eV, or of peaks of any other values, so it cannot be concluded that the NIII is indicative of the ideal condition sequence. Testing this is much more challenging. Testing all permutations of 10 pulses of either 1 or 0 eV, would require something on the order of 2^{10} simulations, and increasing and decreasing the each pulse by only a single increment either way 3^{10}, much less any more peaks/increments.

Comparison of NIII for O_3, O and NO Targeting with Results of Incrementally Increasing or Decreasing Two Successive Peak Values by 0.1 eV

For an initial look at this, a preliminary study of just 2 pulses increased and decreased by a single increment gave a much more manageable number of just 3^2 different condition sequences to simulate (Fig. 8b). Simulations of these 9 different condition sequences were run for three different chemicals, O^3, O and NO. The gas temperature was also varied by ± 20 K coincident with the same time intervals as the electron energy pulses, but these increments were too small to make any significant difference in any of the outcomes, and constraints on time did not permit larger increments to be tried. For this reason, the NIII of gas temperature on the concentrations of O_3, O and NO were not included in this work, and a constant gas temperature of 300 K was used throughout. The NIII's for each target chemical were generated from a "baseline"/default condition sequence of two consecutive 1 eV pulses.

The results of these simulations are shown in Figs. 13, 14, 15 and 16, input (electron energy in this case), productivity (value of targeted output, species concentration of either NO, O and O_3 in this case) and cumulative efficiency (value of targeted output/cumulative input) are shown for each targeted chemical and compared with NIII of electron energy for each species for three depths of predecessor. There was good agreement between the NIII results and the results of the condition sequences for each of the target chemicals. Additionally, the different NIII depths were found to correspond to different time phases in the simulations.

The NIII of electron energy (ε) on ozone (Fig. 13), for example, is (a) negative for the direct precursors (precursor 1/predecessor depth 2) for both electron energy peaks, (b) more strongly negative for the second peak than the first. This suggests that (i) the effect of an increase in either of the two electron energy peaks would result in a decrease of ozone formation (and vice versa for a decrease) and (ii) according to the NIII for depth 2, an increase in the second peak would decrease ozone formation more strongly than an increase in the first. Point (i) is supported given that there is no O_3 formation during the rise of the first peak, and the ozone formation then increases as the first peak falls (Fig. 13). Ozone formation over the course of the second peak is likely to be due to the effects suggested by the NIII's of depths 4 and 6, as there will inevitably be an overlap in the effects of predecessor depths 2, 4, and 6 once O_3 begins to be formed. It seems that as soon as the instantaneous electron energy ceases (at 0.2×10^{-7} s and beyond) the effects "predicted" by depth 4 and 6 predecessors can be immediately observed, which suggests that the effects of NIII's of depth 2 correspond to an instantaneous influence on the target node (in a network of this type).

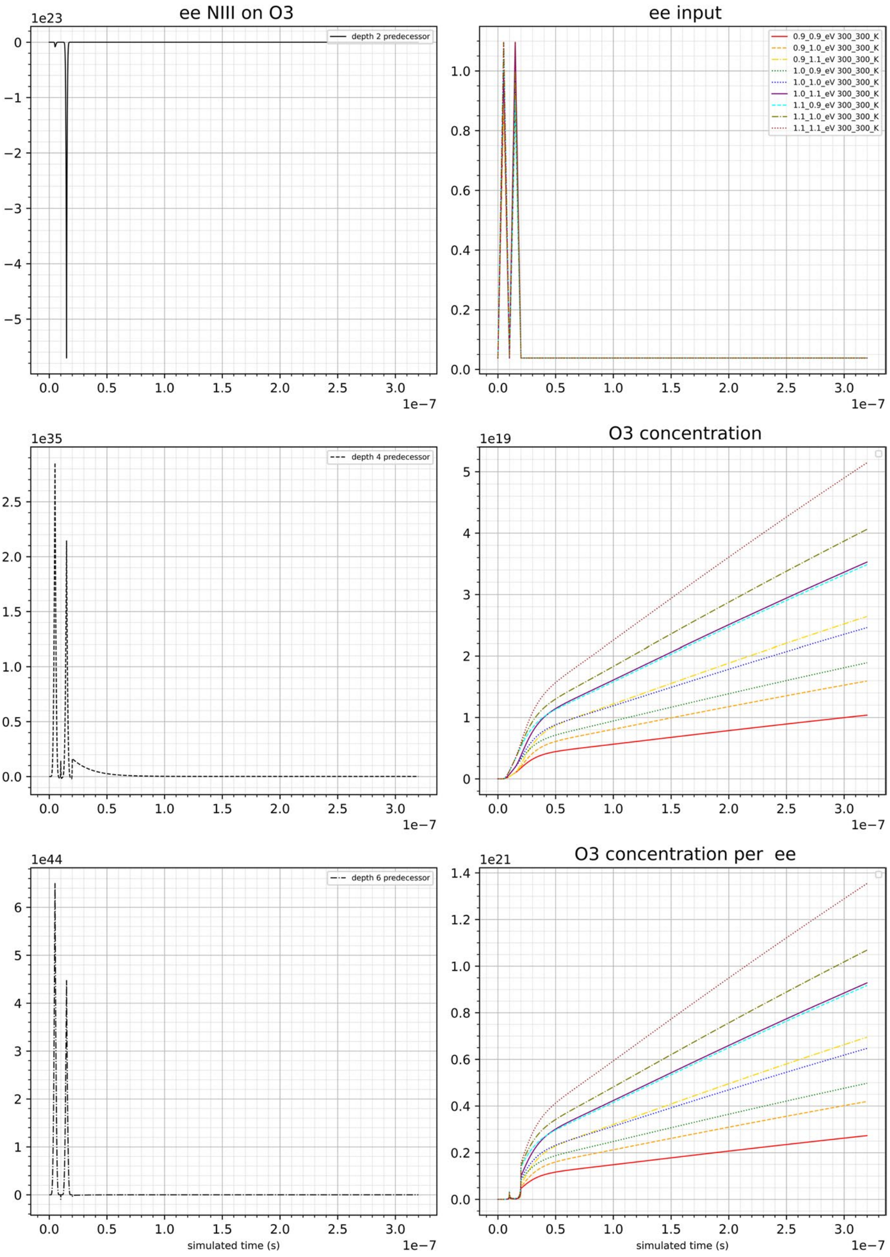

Fig. 13 O_3 (O3) electron energy (ee) pulse configurations. Each colour/linestyle combination represents a specific simulation (as specified by the legend in the top right). The three different depths of predecessor are shown from top (depth 2) to bottom (depth 6)

The depths of 4 and 6 NIII's of electron energy on ozone were both (a) positive for both peaks and (b) more positive for the first peak than the second peak. This suggested that (i) increasing either peak should increase the ozone formation and (ii) increasing the

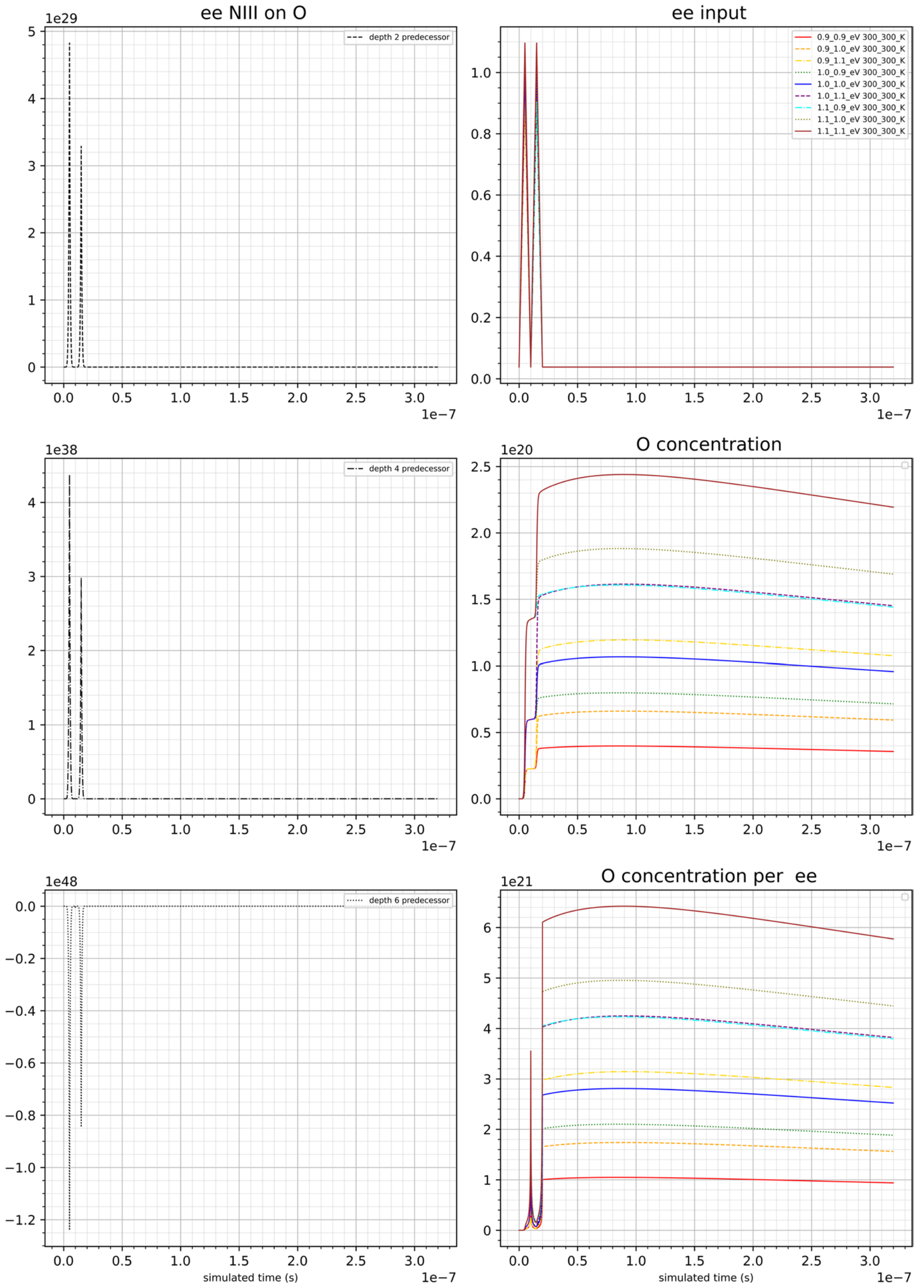

Fig. 14 O electron energy pulse configurations. Each colour/linestyle combination represents a specific simulation (as specified by the legend in the top right). The three different depths of predecessor are shown from top (depth 2) to bottom (depth 6)

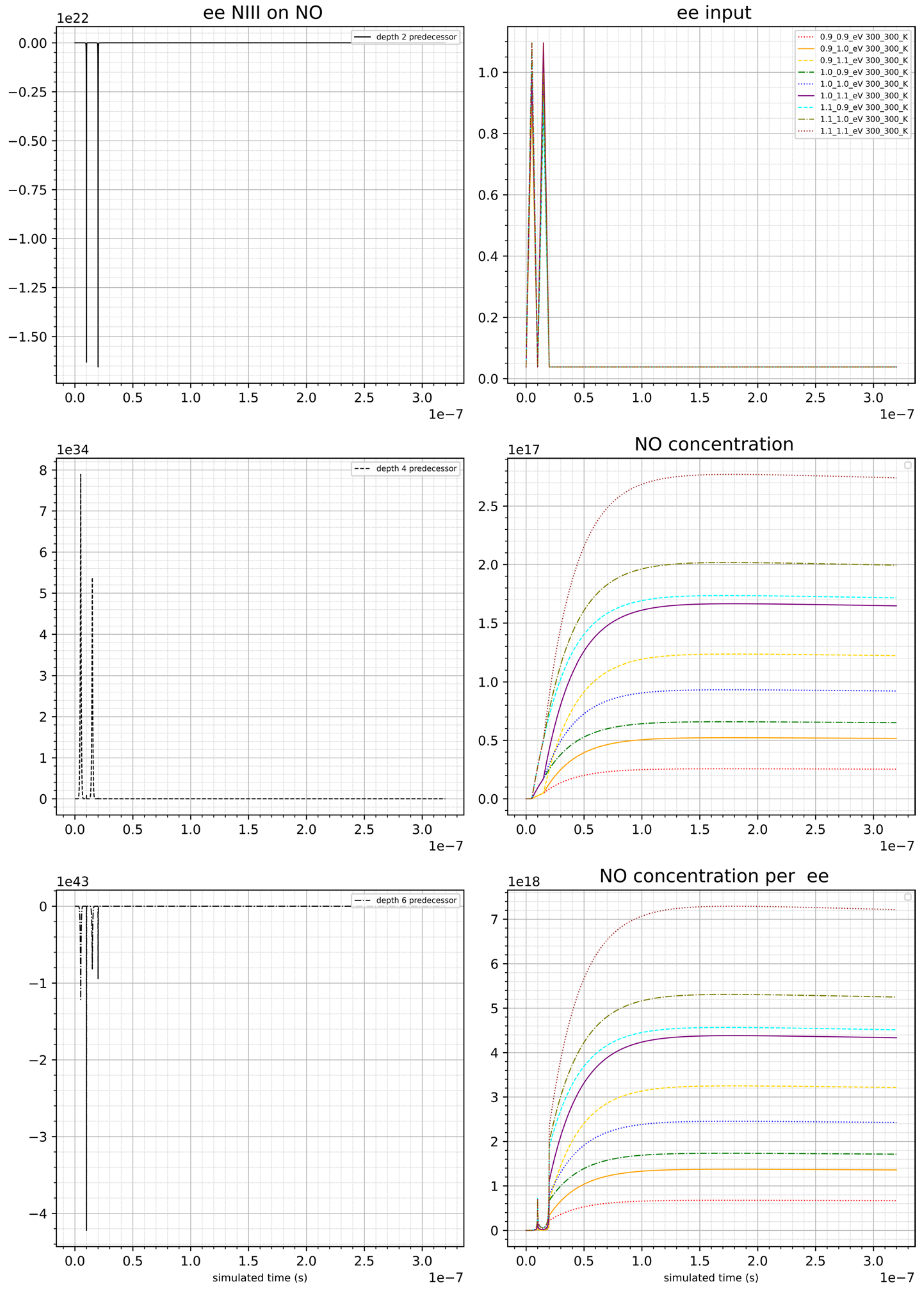

Fig. 15 *NO electron energy pulse configurations. Each colour/linestyle combination represents a specific simulation (as specified by the legend in the top right).* The three different depths of predecessor are shown from top (depth 2) to bottom (depth 6)

first peak should have a stronger effect than increasing the second. Support for point (i) can be observed from the fact that the lowest decrease of both peaks (0.9 eV,0.9 eV) gave the lowest ozone concentrations, and the highest increase of both peaks (1.1 eV,1.1 eV) gave the highest (Fig. 13). Point (ii) can be supported by the observation that, for the same total energy input, a higher peak followed by a lower peak, invariably corresponded to a higher ozone formation than a lower peak followed by a higher peak in every case (compare 1.1 eV,0.9 eV vs 0.9 eV,1.1 eV for example). The fact that both predecessors 4 and 6 are positive corresponds with the continuing increase of O_3, although since predecessor depths 4 and 6 are both positive and have almost identical relative magnitudes in their peaks, it is not possible (at least within the scope of this work) to tell where the effect of one begins and the other ends, and the nature of the overlap between them, though the two effects would likely compound.

Some similar agreement between NIII and condition sequences can be observed for O targeting (Fig. 14). The depth 2 predecessor values are generally positive for NIII of ε on O which seem to correspond to a very strong instantaneous increase in the O concentration. The second increase being a smaller increase than the first, also corresponds to the second NIII "peak" being smaller than the first.

After the initial electron energy pulses, the subsequent trends in O concentration correspond with the NIII results for predecessors 4 and 6. The results of all simulations appear to increase up to approximately 0.8×10^{-7} s, corresponding to a positive depth 4 predecessor for both peaks, and a subsequent decrease for all simulations after $\sim 0.8 \times 10^{-7}$ s corresponds to a negative depth 6 predecessor for both pulses. The ratio of the NIII for the two electron peaks also corresponded with observations from the different simulations, as in a similar manner to the O_3 results, the electron energy pulse has a higher NIII than the second, and the same trend (higher first peak giving a higher target chemical concentration than a higher second peak for the same total energy input) was observed. The fact that depth 6 predecessor results had a more strongly negative first peak than the second peak, suggests that running the simulation for long enough might see these trends reverse. It would indeed be interesting to see if continuing the simulation for longer would result in the O concentration of the 1.1 eV, 1.1 eV simulation eventually decreasing below the 0.9 eV, 0.9 eV simulation, and 1.1 eV, 0.9 eV decreasing below the 0.9 eV, 1.1 eV, but unfortunately this was outside the scope of this study.

The NIII's for NO (Fig. 15) appeared to have two broad peaks that coincided with the two electron energy peaks which appeared to have similar relative magnitudes to the O and O_3 results, and also two very narrow peaks coinciding exactly with the ends of the electron energy peaks. For NIII depth 2, the two narrow peaks had very similar magnitudes, with the first peak having a slightly greater absolute value. The two narrow peaks were absent in NIII depth 4 and present in NIII depth 4 and both were negative with the first peak being between 4 and 5 times the absolute value of the second. The very sharp drops seen in the depth 2 predecessor results of the NIII also appear to correspond with a sharp transition in the concentrations at these points in time.

The broader peaks for NIII depth 4 and 6 appear to show agreement with the simulations in a similar way to O and O_3: increase following the initial electron energy peaks (corresponding with positive depth 4 NIII), then a slight decrease (corresponding with negative depth 6 NIII). There is also the same agreement as O and O_3 with the different combinations of peak heights in the simulations, and the NIII results (higher overall resulting in more NO formation, and higher then lower peaks forming more NO than lower then higher peaks for the same total electron energy). The very steep straight line dips could possibly be due to at least one reaction in the rate coefficients (reaction $e+O_2$ O_2^- [27]) having a step function relationship with two different rate coefficient functions for different electron energy levels. The methods used in this paper currently do not account for this, and future modifications will need to take this possibility into account.

Discussion

Everything in the previous sections suggests a relationship between the NIII and the variation in output resulting from changing the inputs in the simulation over the ranges investigated. Furthermore, the results for O_3, O and NO (Figs. 13, 14 and 15) all suggest that, for the most part, each antecedent graph-theoretical depth from the target node appears to correspond with each subsequent temporal phase in the simulation, resolving the issue of the contradictory NIII results for the different depths mentioned in previous sections. This, and the correlation of the results in Figs. 11 and 12 suggest that the OCARINA method can be effective for targeted condition sequencing for these 0D time-dependent study. More broadly, the results suggest that more locally optimal conditions for targeted chemical outcomes can be identified by the OCARINA method of analysis of the relationship between variables in the context of a network. This has been achieved by a variable-relationship bipartite graph such as the one described in Figs. 1, 2, 3, 4, 5 which, once made, can be evaluated at any point in a time-dependent simulation by a simple matrix operation on the graph adjacency matrix.

The OCARINA method should even be scalable to multiple spatial elements, scaling the method up to 1D and higher, provided this can be represented by a variable-relationship graph. With finite element methods, for example, once a finite element array is defined the numerical approximation to the PDE (Partial Differential Equation) system is a network of nodal ODEs (Ordinary Differential Equations) [31]. The variable relationship graphs for this would be much larger than those used in this work, as an increase in the number of spatial elements would mean an exponential increase in the number of elements in the adjacency matrix of the graph, and an increase in the number of dimensions often means an exponential increase in the number of spatial elements. Fortunately, the simulations would also scale in a similar manner, thus the OCARINA method should scale in the same way as the model it is used on. Given that the main components of this analysis are matrix operations, the OCARINA method could also be shared across multiple processors if the application needs to be scaled up for more complex systems.

Since the Katz centrality also operates on the adjacency matrix, this would also scale accordingly. The analysis of the results in Fig. 10 suggest that the Katz centrality gives a measure of the relative influence of a node (in this case electron energy) over time with respect to the whole graph. Although the Katz centrality is described as a solution to eigenvector centrality for directed graphs, there were still issues with its use. As discussed in [32], there is not a strict definition of what makes a centrality measure, and researchers are constantly proposing variations to existing centrality measures. However, Lawyer [33] suggests that while centrality measures are good at highlighting the very most influential nodes, they do not do a good job of quantifying the importance of the other nodes (mid-importance nodes for lack of a better phrase). Identifying the most influential node in a network is useful, as in the absence of any other information, this might be the node that is most worth attention. Nevertheless, it is very unlikely that this node will have equal influence on all the other nodes, and may have little influence on a particular target node. Thus Katz centrality values on their own are not sufficient for targeted condition sequencing.

Limitations of the Methods in this Study

Whilst the OCARINA should be scalable to be applied to more complex models and higher dimensions, the difficulties of verifying the OCARINA algorithm results also scale accordingly. To be as rigorous as possible would require all possible condition sequences to be investigated and compared to the output of the OCARINA algorithm. Ideally a parametric sweep of a range of different values of all ten pulses would have been simulated, but even as little as ten values of each pulse (0.1 eV, 0.2 eV, 0.3 eV … 1 eV, for example) would result in 10^{10} combinations, each one needing a separate simulation. Unfortunately, this was computationally unfeasible within the timescales of the project and the available computing power. For this same reason, NIII's of multiple variables were not looked into (except for the omitted gas temperature results mentioned above).

Another limitation of the study was that the main application of the algorithm was on the electron energy. Whilst this is clearly the right variable to focus on for plasma chemical waveform targeting, it is also a variable with no predecessors. This study is therefore limited to observing the effects of altering completely independent variables, and not variables which are embedded in the cycles of the network (such as most of the reactions and species) and thus are both influenced by as well as influencing the target node. This also includes the target node itself.

Some additional issues also need to be discussed. Looking at Figs. 1, 2, 3, 4, 5, 6, and 7, one might wonder how increasing one of the variables might inadvertently affect the others. Clearly the weight of each variable to reaction edge depends on more than just the start and end nodes, they are functions of the other variables in the relationship. In Fig. 3, for example, changing the electron energy would change other things outside of this reaction that would indirectly change e and O2 concentrations, and this would alter the effect that increasing the electron energy would have. This might mean the very means of defining the effect of increasing the electron energy (NIII), change as soon as it's increased. Whilst this is true, this is why the term "Net Influence of Infinitesimal Increase is accurate, as it is only strictly true if every variable is changed infinitesimally. As mentioned previously, applying the OCARINA method to the dynamic graph works under the assumption that an increase in small enough increments will be a close approximation to the infinitesimal increase. Also, as shown in Figs. 13, 14 and 15, the effects of a node on all other nodes and on itself is effectively increasingly taken into account as the node depths of analysis increase. Each depth of analysis is done on the whole adjacency matrix, including the target node. The results in Figs. 13, 14 and 15 also suggest that indirect "feedback" effects of a node on itself are not instantaneous, as this is a time dependent simulation, only a certain amount of change can happen in a given time step. This being the case, it would be interesting to include 'time' itself as a node in these variable relationship graphs in future.

Estimating the Increments

Figure 12 shows that it is possible in some cases to make inferences outside the small ranges of the small variable space covered by the 9 different conditional sequences in Fig. 8b, and whilst results like this cannot be guaranteed, it suggests there are scenarios in which changes can be made in greater increments than those shown in Figs. 8b and 13, 14, 15 and 16. It would be convenient if these scenarios could be identified, and it may be possible to do this using VR range graphs, but this will have to be investigated in future studies.

Being as precise as possible, the y-axis NIII plots ought to be labelled “Predicted net relative change in the value of the target node as a result of an infinitesimally small positive change in the value of the given source node at its current value”. For a dynamic graph, all these numbers are true only at the immediate values of the source nodes. Knowing how much to change the inputs by is key to being able to make further use of this novel method. If the NIII results indicate that increasing an input would be strongly conducive to increasing an output, should the input be increased by 10%? 20%? 50%? Etc.? What if increasing it gives the opposite result?

The results of the NIII are true for an infinitesimally small increase, essentially a single point. The closer a real increase is to this single point, the closer the result of the increase will be to what the NIII suggests it will be. Thus the smaller the increase the better, and an algorithm that moves over very small steps will likely be the most accurate. This will of course be more time-consuming, so increasing the step sizes in changes in the variable will be desirable. However, too great an increase may overshoot the peak in the relationship and begin a decrease, and so identifying where the peaks are is the missing ingredient to enhancing this method. Further to the previous point, the NIII is composed of a large number of derivatives, which are all only true for single points along the graphs of each relationship, and the distance from each point to a local maximum (peak) of each relationship must be different. It is possible to know where each differential is on the x and y axis in each relationship, and therefore how far away from the local peak that is. Local maxima and minima of each relationship can be found before any simulations are run and stored as edge attributes, and the difference between those and the point of the NIII for each relationship could be measured at any point during a simulation. This could be done for each edge and the resulting values used for an error measurement for the NIII. Local maxima and minima will vary depending on the values of all the independent variables for each relationship, so 2, 3 or higher dimensional shapes might be needed to represent this error (which could be represented as an n-dimensional identity matrix, for example), which again could be calculated before the simulation is run. This could point the way to assessing how large an incremental increase or decrease should be. Finally, it must be noted on making incremental changes that if a variable is changed earlier on in the simulation, this will effectively invalidate the NIII readings after that point. It may be best to make changes to the last time steps first.

Potential Variations of Set 0: Multiple Targets, Variable Targets and Anti-Targets

Looking at Set 0

$$Set0 = A_0, B_0, C_0, D_0, E_0, T_{e0}$$

The only value of this set used in the example was $D_0 = 1$, all the other variable values = 0. Clearly, if D_0 was set to -1, and the same method run, then the output of the algorithm would be conducive to reducing the value of D_0. Hence D_0 would be an “anti-target”. This being the case, the algorithm could be run with one target, and one anti-target. Say, for example, we want to avoid a troublesome by-product of a process, we could set this as an anti-target. If there are any pathways that favour the target without the anti-target, the NIII will be composed of these, as any pathways shared by the target and the anti-target will be cancelled out.

It also follows that more than one variable can be set as the target (if two chemicals are required for example) and that they could be given different set 0 values if there is an inclination to produce them in different ratios. Clearly there will be many scenarios where this will not be possible due to the number of pathways shared by the multiple targets/anti-targets. A preliminary graph analysis could be run to assess the number of shared relationships/pathways to inform whether such a multi-target analysis would be worth pursuing.

How this Translates to Systems in Reality

Using these graph theory methods are only as good as the models that the graphs are formed from. Anything that applies to the model, applies to the graph. If the models do not accurately represent reality, then the graphs will not either. If big assumptions are made to simplify the model, these will apply to the graph as well. Assuming the model is an accurate representation of a system in reality, the results of this algorithm highlight changes which can be made to the variables, which correspond to actions in the real system.

These results are based on simple electron energy pulses aimed to roughly simulate streamers in a DBD plasma source. The relationship between the voltage waveform for this plasma source and the DBD streamers is not a direct one and is dependent on the frequency and the plasma source itself. The methods described in this paper, however, could be applied to a mathematical model that accurately represents the relationships between the power supply, voltage waveform, plasma source and DBD streamers, assuming such a model is practical. In the absence of such a model, the NIII of the electron energy could be used to identify the ideal sequence of electron energy magnitudes, which in practice could be supplied either successively in time, or successively in space (e.g. if a gas flow is moving through successive discharge conditions). Aside from electron energy and gas temperature, if a reaction relationship, for example, has a high NIII, an action would be to somehow catalyse that reaction. If a chemical variable, for example, has a high NIII, an action could be to inject some of that chemical into the system at the time the NIII is highest.

Wider Applicability

More generally the question of how much electron energy to supply is a question of when to supply any input for a given output. Put in the simplest terms, the question of “how much of what to supply and when?” is an almost universal one. The answer to this is “when it matters most”, and something could be said to matter most at the point it has the strongest influence on an intended outcome, i.e. where it will make the biggest difference with a capital Δ.

For any system that can be adequately described by a series of time-varying equations, a variable-relationship graph can be made by forming edges and nodes from each of its equations as above to form a weighted directional bipartite graph containing both positive and negative edges. As a great number of systems can be described by such equations, the techniques outlined in this work could have much wider applicability not just in reaction engineering but even in engineering in general. A general application of this work may therefore be a useful addition to the toolkit of techniques for improving the processes we all depend on.

Conclusions

It has been demonstrated that a plasma chemical reaction system, including electron energy and gas temperature, can be represented as a dynamic bipartite variable-relationship (VR) graph. The results suggest that the OCARINA method of analysis can be used to identify more effective conditional sequencing of electron energy for a target chemical species.

Using this dynamic VR graph, the OCARINA method described in [15] can be used at any point on a time-dependent 0D chemical kinetic simulation, and can also be simplified to a vector–matrix product of the VR graph adjacency matrix with a target identity vector. This matrix operation can be repeated for successive 'depths' of predecessors of the target node without any loss of computational expediency. The resulting set of values give a measure of the 'net influence' of increasing the value of each node on a target node, which are different at different depths.

0D chemical kinetic simulations were run with 1–10 electron energy pulses and it was observed that each successive pulse gave diminishing returns on ozone output. This agreed with the 'net influence of incremental increase' (NIII) results of running the OCARINA algorithm for a baseline of continuous pulses. The eigenvector centrality index (ECI) and Katz centrality index (KCI) were also run the graph for this same baseline simulation and it was found that the ECI was not suitable for this type of graph, but the reverse KCI gave meaningful values. 0D chemical kinetic simulations were also run for each combination of $\pm 10\%$ of the peak values of just two electron energy pulses. OCARINA was run on a baseline of the default values of these two pulses. The results of these kinetic simulations largely agreed with the NIII results of the OCARINA algorithm for target nodes of O_3, NO and O. It was also observed that the NIII for each antecedent depth of predecessor, appeared to correspond with a successive temporal 'phase' in each simulation.

Simple alterations to the target identity vector could also allow OCARINA to be run on other target species, and even on multiple target chemical species and also negative target nodes, though this will need to be examined in future work. These methods could also potentially scale to 1D, 2D and higher dimensional models, and could potentially be applied to any time dependent set of equations, provided they can be represented with a VR graph.

Future work should therefore include investigating the effects of altering variables other than the electron energy, in finding the limits of how these variables can be increased in response to the NIII results (i.e. to more rapidly approach optimal conditions) and also in applying these methods to other systems and higher dimensional models. Further development of these methods could be a key component of a dynamical systems alternative to the steady state, mixed-integer nonlinear optimisation method.

Appendix

Here the steps are outlined for the generalisation of the logical statements in the OCARINA method [15] to matrix operations on variable-relationship graphs (Figs. 1, 2, 3 and 4) to verify that the two are essentially equivalent, and that the matrix operations can therefore be applied as a node-influence ranking method.

To begin, a *target node* (*X*) must be specified in the variable relationship graph (in this case a variable node) which represents the outcome we wish to change (if we are targeting ozone for example, the target node would be the O_3 variable node). Given that any node in a directional graph can only be directly influenced by a node directly preceding it, we need only be concerned with predecessors. Given also that a variable is always preceded by a single relationship, which in turn is always preceded by one or more variables (unless the variable is a completely independent variable), any node affects the target node through a series of successor nodes as follows:

$$V_n \rightarrow R_n \cdots \rightarrow V_2 \rightarrow R_{ii} \rightarrow V_1 \rightarrow R_i \rightarrow X$$

The total of all such pathways between any node and X constitute the total/net influence of this node on the target node X.

Clearly a node could be present at multiple points (depths) along a single chain and be present at different depths along different chains. It therefore follows that the node influence will be different at different depths away from the target node (X). This being the case it is important to be able to evaluate the net-influence of each node at different depths. This can be done using the logic in Tables 2, 3 and 4, and using the relationships in the following sections.

The influence of a direct predecessor *relationship* (R_i) on the target node X can be assessed using the logic in Table 2. In turn, the influence of a variable V_1 on X can be assessed by assessing its influence on R_i, given R_i ‘s influence on X, if V_1 is a direct predecessor of R_i (Table 3). Going a further step back through the predecessors in the graph, this logic can then be applied again as shown in Table 4, and so on. In such a graph, the simplest measure of the "net influence" of any node on the target node, is the sum of *all* of its pathways of influence on the target node X (*all positive pathways minus all negative pathways*) at a given depth.

Nodes must be grouped into one or more sets depending on if they satisfy one or more of these criteria. Set O is the set containing only the target node. All other sets are sets of predecessors to the target node. Each set is named according to how its elements affect their successors. *Set PN*, for example, contains only nodes which *positively* affect nodes which *negatively* affect the target node, whereas *set PNPP* contains only nodes which *positively* affect nodes which *negatively* affect nodes which *positively* affect nodes which *positively* affect the target node.

Each node in each set is assigned a value to measure the extent of its effect on the node X. The same node can be present in more than one set, but its value calculation will be specific to that set. These values of each node (species or reaction) in each set are evaluated using the simplest operations possible, multiplication and addition, as follows:

$$Value\ of\ R\ in\ setP = \left(\sum each\ positive\ edge\ from\ R\ to\ X\right)(Xvalue)$$

Table 2 Conditional statements for ranking direct (depth 1) predecessors of a target node X

Conditional statement	Delineation	Category	Set
If a relationship ($\boldsymbol{R_i}$) positively affects $\boldsymbol{X}$, this relationship is desirable	$\boldsymbol{R_i}$ -p->$\boldsymbol{X}$	Desirable	P
If a relationship ($\boldsymbol{R_i}$) negatively affects $\boldsymbol{X}$, this relationship is undesirable	$\boldsymbol{R_i}$ -n->$\boldsymbol{X}$	Undesirable	N

Table 3 Conditional statements for ranking depth 2 predecessors of a target node X

Conditional statement	Delineation	Category	Set
If a relationship ($\boldsymbol{R_i}$) positively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) positively affects this relationship, this variable ($\boldsymbol{V_1}$) is desirable	$\boldsymbol{V_1}$ -p->$\boldsymbol{R_i}$ -p->$\boldsymbol{X}$	Desirable	PP
If a relationship ($\boldsymbol{R_i}$) positively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) negatively affects this relationship, this variable ($\boldsymbol{V_1}$) is undesirable	$\boldsymbol{V_1}$ -n->$\boldsymbol{R_i}$ -p->$\boldsymbol{X}$	Undesirable	NP
If a relationship ($\boldsymbol{R_i}$) negatively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) positively affects this relationship, this variable ($\boldsymbol{V_1}$) is undesirable	$\boldsymbol{V_1}$ -p->$\boldsymbol{R_i}$ -n->$\boldsymbol{X}$	Desirable	PN
If a relationship ($\boldsymbol{R_i}$) negatively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) negatively affects this relationship, this variable ($\boldsymbol{V_1}$) is desirable	$\boldsymbol{V_1}$ -n->$\boldsymbol{R_i}$ -n->$\boldsymbol{X}$	Undesirable	NN

Table 4 Conditional statements for ranking depth 3 predecessors of a target node X

Conditional statement	Delineation	Category	Set
If a relationship ($\boldsymbol{R_i}$) positively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) positively affects this relationship, and a preceding relationship ($\boldsymbol{R_{ii}}$) positively affects this variable ($\boldsymbol{V_1}$), this relationship ($\boldsymbol{R_{ii}}$) is desirable	$\boldsymbol{R_{ii}}$ -p- > $\boldsymbol{V_1}$ -p- > $\boldsymbol{R_i}$ -p- > $\boldsymbol{X}$	Desirable	PPP
If a relationship ($\boldsymbol{R_i}$) positively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) positively affects this relationship, and a preceding relationship ($\boldsymbol{R_{ii}}$) negatively affects this variable ($\boldsymbol{V_1}$), this relationship ($\boldsymbol{R_{ii}}$) is undesirable	$\boldsymbol{R_{ii}}$ -n- > $\boldsymbol{V_1}$ -p- > $\boldsymbol{R_i}$ -p- > $\boldsymbol{X}$	Undesirable	NPP
If a relationship ($\boldsymbol{R_i}$) positively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) negatively affects this relationship, and a preceding relationship ($\boldsymbol{R_{ii}}$) positively affects this variable ($\boldsymbol{V_1}$), this relationship ($\boldsymbol{R_{ii}}$) is undesirable	$\boldsymbol{R_{ii}}$ -p- > $\boldsymbol{V_1}$ -n- > $\boldsymbol{R_i}$ -p- > $\boldsymbol{X}$	Undesirable	PNP
If a relationship ($\boldsymbol{R_i}$) positively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) negatively affects this relationship, and a preceding relationship ($\boldsymbol{R_{ii}}$) negatively affects this variable ($\boldsymbol{V_1}$), this relationship ($\boldsymbol{R_{ii}}$) is desirable	$\boldsymbol{R_{ii}}$ -n- > $\boldsymbol{V_1}$ -n- > $\boldsymbol{R_i}$ -p- > $\boldsymbol{X}$	Desirable	NNP
If a relationship ($\boldsymbol{R_i}$) negatively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) positively affects this relationship, and a preceding relationship ($\boldsymbol{R_{ii}}$) positively affects this variable ($\boldsymbol{V_1}$), this relationship ($\boldsymbol{R_{ii}}$) is undesirable	$\boldsymbol{R_{ii}}$ -p- > $\boldsymbol{V_1}$ -p- > $\boldsymbol{R_i}$ -n- > $\boldsymbol{X}$	Undesirable	PPN
If a relationship ($\boldsymbol{R_i}$) negatively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) positively affects this relationship, and a preceding relationship ($\boldsymbol{R_{ii}}$) negatively affects this variable ($\boldsymbol{V_1}$), this relationship ($\boldsymbol{R_{ii}}$) is desirable	$\boldsymbol{R_{ii}}$ -n- > $\boldsymbol{V_1}$ -p- > $\boldsymbol{R_i}$ -n- > $\boldsymbol{X}$	Desirable	NPN
If a relationship ($\boldsymbol{R_i}$) negatively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) negatively affects this relationship, and a preceding relationship ($\boldsymbol{R_{ii}}$) positively affects this variable ($\boldsymbol{V_1}$), this relationship ($\boldsymbol{R_{ii}}$) is desirable	$\boldsymbol{R_{ii}}$ -p- > $\boldsymbol{V_1}$ -n- > $\boldsymbol{R_i}$ -n- > $\boldsymbol{X}$	Desirable	PNN
If a relationship ($\boldsymbol{R_i}$) negatively affects $\boldsymbol{X}$, and a variable ($\boldsymbol{V_1}$) negatively affects this relationship, and a preceding relationship ($\boldsymbol{R_{ii}}$) negatively affects this variable ($\boldsymbol{V_1}$), this relationship ($\boldsymbol{R_{ii}}$) is undesirable	$\boldsymbol{R_{ii}}$ -n- > $\boldsymbol{V_1}$ -n- > $\boldsymbol{R_i}$ -n- > $\boldsymbol{X}$	Undesirable	NNN

$$Value\ of\ R\ in\ set\ N = \left(\sum\ each\ negative\ edge\ from\ R\ to\ X\right)(Xvalue)$$

As every value in every set is effectively a multiple of the value given to X in set 0, the default value of X in set 0 is 1.

$$Value\ of\ V\ in\ set\ PP = \left(\sum\ all\ positive\ edges\ from\ V\ to\ R\ in\ set\ P\right)(value\ of\ R\ in\ set\ P)$$

$$Value\ of\ V\ in\ set\ NP = \left(\sum\ all\ negative\ edges\ from\ V\ to\ R\ in\ set\ P\right)(value\ of\ R\ in\ set\ P)$$

$$Value\ of\ V\ in\ set\ PN = \left(\sum\ all\ positive\ edges\ from\ V\ to\ R\ in\ set\ N\right)(value\ of\ R\ in\ set\ N)$$

$$Value\ of\ V\ in\ set\ N = \left(\sum\ all\ negative\ edges\ from\ V\ to\ R\ in\ set\ N\right)(value\ of\ R\ in\ set\ N)$$

The *"desirability"* or *"demand"* of a node (for increasing the value of the target node) is the sum of all the parallel values of a node at the specified number of predecessor steps from the target node. A variable node (V) for example, would have its "desirability" ranked/scored as follows:

$$V \in\ Final\ ranking\ set =$$

$$V \in PPPP + V \in NPPP + V \in PNPP + V \in NNPP$$

$$+V \in PPNP + V \in NPNP + V \in PNNP + V \in NNNP$$

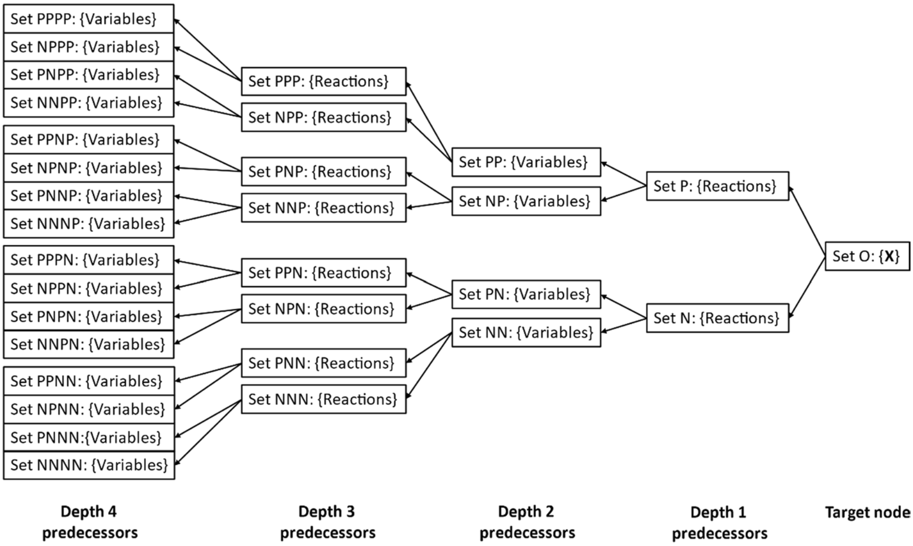

Fig. 16 A map of the logic of the predecessor ranking, the predecessors are ranked according to how they influence a specified target node. This figure also serves to illustrate the problem of computing this logically, that as more precursive stages are introduced, the number of sets, and therefore the number of computations, effectively doubles

$$+V \in PPPN + V \in NPPN + V \in PNPN + V \in NNPN$$

$$+V \in PPNN + V \in NPNN + V \in PNNN + V \in NNNN$$

Nodes with net positive or negative ranks should result in an overall increase or decrease in the value of the target node respectively.

The dependencies of these calculations on the sets are delineated in Fig. 16. It is clear from Fig. 16 that with each additional depth of predecessors to the sequence, the number of sets that would need to be processed doubles. This would lead to predecessor ranking becoming computationally unfeasible this way if the number of predecessors is too great.

This raises the question: If the process was done by adding the sets together at each depth before evaluating the next depth, instead of evaluating a negative and positive set for each set in the succeeding depth, would this produce the same results? Or would some pathways of influence cancel each other out if added together, which would eliminate effects at preceding depths being taken into account?

Conveniently, it can be shown that adding the sets together at any depth to make a single set, and evaluating preceding sets starting from that, is exactly equivalent to the above process, and thus won't change the outcome at any depth. The following section verifies this.

Reduction of the set summation operations to adjacency matrix operations.

Using the graph in Fig. 4 as an example, since *D* is the target node, we set the value of *D* as *1*, and all the other values as 0 in set 0.

Target node set:

$$Set0 = A_0, B_0, C_0, D_0, E_0, T_{e0}$$

Each node "score" in set P and set N are then functions of the values in set 0, as follows:

Depth 1 predecessors:

$$SetP = R_{1P}, R_{2P}, R_{3P}$$

$$SetN = R_{1N}, R_{2N}, R_{3N}$$

$$R_{11} = R_{1P} + R_{1N} = \left(\left(v_5 \cdot B_0\right) + \left(v_1 \cdot C_0\right)\right) + \left(v_2 \cdot A_0\right)$$

$$R_{21} = R_{2P} + R_{2N} = \left(v_6 \cdot E_0\right) + (\left(v_7 \cdot B_0\right) + \left(v_4 \cdot C_0\right))$$

$$R_{31} = R_{3P} + R_{3N} = ((v_{10} \cdot D_0) + \left(v_3 \cdot A_0\right)) + ((v_8 \cdot A_0) + \left(v_9 \cdot B_0\right))$$

Each node "score" in set PP, NP, PN and NN are then functions of the values in sets P and N as follows:

Depth 2 predecessors:

$$SetPP = A_{PP}, B_{PP}, C_{PP}, D_{PP}, E_{PP}, T_{ePP}$$

$$SetNP = A_{NP}, B_{NP}, C_{NP}, D_{NP}, E_{NP}, T_{eNP}$$

$$SetPN = A_{PN}, B_{PN}, C_{PN}, D_{PN}, E_{PN}, T_{ePN}$$

$$SetNN = A_{NN}, B_{NN}, C_{NN}, D_{NN}, E_{NN}, T_{eNN}$$

$$\begin{aligned} A_2 =& A_{PP} + A_{NP} + A_{PN} + A_{NN} = ((r_3 \cdot R_{1P}) + (0 \cdot R_{2P}) + (r_6 \cdot R_{3P})) + ((r_3 \cdot R_{1P}) + (0 \cdot R_{2P}) \\ &+ (r_6 \cdot R_{3P})) + ((r_3 \cdot R_{1N}) + (0 \cdot R_{2N}) + (r_6 \cdot R_{3N})) + ((r_3 \cdot R_{1N}) + (0 \cdot R_{2N}) + (r_6 \cdot R_{3N})) \end{aligned}$$

$$\begin{aligned} B_2 =& B_{PP} + B_{NP} + B_{PN} + B_{NN} = ((0 \cdot R_{1P}) + (r_5 \cdot R_{2P}) + (r_7 \cdot R_{3P})) + ((0 \cdot R_{1P}) + (r_5 \cdot R_{2P}) \\ &+ (r_7 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (r_5 \cdot R_{2N}) + (r_7 \cdot R_{3N})) + ((0 \cdot R_{1N}) + (r_5 \cdot R_{2N}) + (r_7 \cdot R_{3N})) \end{aligned}$$

$$\begin{aligned} C_2 =& C_{PP} + C_{NP} + C_{PN} + C_{NN} = ((0 \cdot R_{1P}) + (r_4 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1P}) + (r_4 \cdot R_{2P}) \\ &+ (0 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (r_4 \cdot R_{2N}) + (0 \cdot R_{3N})) + ((0 \cdot R_{1N}) + (r_4 \cdot R_{2N}) + (0 \cdot R_{3N})) \end{aligned}$$

$$\begin{aligned} D_2 =& D_{PP} + D_{NP} + D_{PN} + D_{NN} = ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) \\ &+ (0 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N})) \end{aligned}$$

$$\begin{aligned} E_2 =& E_{PP} + E_{NP} + E_{PN} + E_{NN} = ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) \\ &+ (0 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N})) \end{aligned}$$

$$\begin{aligned} T_{e2} =& T_{ePP} + T_{eNP} + T_{ePN} + T_{eNN} = ((r_1 \cdot R_{1P}) + (r_2 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((r_1 \cdot R_{1P}) + (r_2 \cdot R_{2P}) \\ &+ (0 \cdot R_{3P})) + ((r_1 \cdot R_{1N}) + (r_2 \cdot R_{2N}) + (0 \cdot R_{3N})) + ((r_1 \cdot R_{1N}) + (r_2 \cdot R_{2N}) + (0 \cdot R_{3N})) \end{aligned}$$

And for all sets in the next stage:
Depth 3 predecessors:

$$SetPPP = R_{1PPP}, R_{2PPP}, R_{3PPP}$$

$$SetNPP = R_{1NPP}, R_{2NPP}, R_{3NPP}$$

$$SetPNP = R_{1PNP}, R_{2PNP}, R_{3PNP}$$

$$SetNNP = R_{1NNP}, R_{2NNP}, R_{3NNP}$$

$$SetPPN = R_{1PPN}, R_{2PPN}, R_{3PPN}$$

$$SetNPN = R_{1NPN}, R_{2NPN}, R_{3NPN}$$

$$SetPNN = R_{1PNN}, R_{2PNN}, R_{3PNN}$$

$$SetNNN = R_{1NNN}, R_{2NNN}, R_{3NNN}$$

Relationships:

$$\begin{aligned}R_{13} &= R_{1PPP} + R_{1NPP} + R_{1PNP} + R_{1NNP} + R_{1PPN} + R_{1NPN} + R_{1PNN} + R_{1NNN}\\ &= \left(\left(v_5 \cdot B_{PP}\right) + \left(v_1 \cdot C_{PP}\right)\right) + \left(v_2 \cdot A_{PP}\right) + \left(\left(v_5 \cdot B_{NP}\right) + \left(v_1 \cdot C_{NP}\right)\right)\\ &\quad + \left(v_2 \cdot A_{NP}\right) + \left(\left(v_5 \cdot B_{PN}\right) + \left(v_1 \cdot C_{PN}\right)\right) + \left(v_2 \cdot A_{PN}\right)\\ &\quad + \left(\left(v_5 \cdot B_{NN}\right) + \left(v_1 \cdot C_{NN}\right)\right) + \left(v_2 \cdot A_{NN}\right)\end{aligned}$$

$$\begin{aligned}R_{23} &= R_{2PPP} + R_{2NPP} + R_{2PNP} + R_{2NNP} + R_{2PPN} + R_{2NPN} + R_{2PNN} + R_{2NNN}\\ &= \left(v_6 \cdot E_{PP}\right) + \left(\left(v_7 \cdot B_{PP}\right) + \left(v_4 \cdot C_{PP}\right)\right) + \left(v_6 \cdot E_{NP}\right) + \left(\left(v_7 \cdot B_{NP}\right) + \left(v_4 \cdot C_{NP}\right)\right)\\ &\quad + \left(v_6 \cdot E_{PN}\right) + \left(\left(v_7 \cdot B_{PN}\right) + \left(v_4 \cdot C_{PN}\right)\right) + \left(v_6 \cdot E_{NN}\right) + (\left(v_7 \cdot B_{NN}\right) + \left(v_4 \cdot C_{NN}\right))\end{aligned}$$

$$\begin{aligned}R_{33} &= R_{3PPP} + R_{3NPP} + R_{3PNP} + R_{3NNP} + R_{3PPN} + R_{3NPN} + R_{3PNN} + R_{3NNN}\\ &= (\left(v_{10} \cdot D_{PP}\right) + \left(v_3 \cdot A_{PP}\right)) + \left(\left(v_8 \cdot A_{PP}\right) + \left(v_9 \cdot B_{PP}\right)\right) + (\left(v_{10} \cdot D_{NP}\right)\\ &\quad + \left(v_3 \cdot A_{NP}\right)) + \left(\left(v_8 \cdot A_{NP}\right) + \left(v_9 \cdot B_{NP}\right)\right) + (\left(v_{10} \cdot D_{PN}\right) + \left(v_3 \cdot A_{PN}\right))\\ &\quad + \left(\left(v_8 \cdot A_{PN}\right) + \left(v_9 \cdot B_{PN}\right)\right) + (\left(v_{10} \cdot D_{NN}\right) + \left(v_3 \cdot A_{NN}\right))\\ &\quad + (\left(v_8 \cdot A_{NN}\right) + \left(v_9 \cdot B_{NN}\right))\end{aligned}$$

And for all sets in the next stage:
Depth 4 predecessors:

$$SetPPPP = A_{PPPP}, B_{PPPP}, C_{PPPP}, D_{PPPP}, E_{PPPP}, T_{ePPPP}$$

$$SetNPPP = A_{NPPP}, B_{NPPP}, C_{NPPP}, D_{NPPP}, E_{NPPP}, T_{eNPPP}$$

$$SetPNPP = A_{PNPP}, B_{PNPP}, C_{PNPP}, D_{PNPP}, E_{PNPP}, T_{ePNPP}$$

$$SetNNPP = A_{NNPP}, B_{NNPP}, C_{NNPP}, D_{NNPP}, E_{NNPP}, T_{eNNPP}$$

$$SetPPNP = A_{PPNP}, B_{PPNP}, C_{PPNP}, D_{PPNP}, E_{PPNP}, T_{ePPNP}$$

$$SetNPNP = A_{NPNP}, B_{NPNP}, C_{NPNP}, D_{NPNP}, E_{NPNP}, T_{eNPNP}$$

$$SetPNNP = A_{PNNP}, B_{PNNP}, C_{PNNP}, D_{PNNP}, E_{PNNP}, T_{ePNNP}$$

$$SetNNNP = A_{NNNP}, B_{NNNP}, C_{NNNP}, D_{NNNP}, E_{NNNP}, T_{eNNNP}$$

$$SetPPPN = A_{PPPN}, B_{PPPN}, C_{PPPN}, D_{PPPN}, E_{PPPN}, T_{ePPPN}$$

$$SetNPPN = A_{NPPN}, B_{NPPN}, C_{NPPN}, D_{NPPN}, E_{NPPN}, T_{eNPPN}$$

$$SetPNPN = A_{PNPN}, B_{PNPN}, C_{PNPN}, D_{PNPN}, E_{PNPN}, T_{ePNPN}$$

$$SetNNPN = A_{NNPN}, B_{NNPN}, C_{NNPN}, D_{NNPN}, E_{NNPN}, T_{eNNPN}$$

$$SetPPNN = A_{PPNN}, B_{PPNN}, C_{PPNN}, D_{PPNN}, E_{PPNN}, T_{ePPNN}$$

$$SetNPNN = A_{NPNN}, B_{NPNN}, C_{NPNN}, D_{NPNN}, E_{NPNN}, T_{eNPNN}$$

$$SetPNNN = A_{PNNN}, B_{PNNN}, C_{PNNN}, D_{PNNN}, E_{PNNN}, T_{ePNNN}$$

$$SetNNNN = A_{NNNN}, B_{NNNN}, C_{NNNN}, D_{NNNN}, E_{NNNN}, T_{eNNNN}$$

Variables:

$$\begin{aligned} A_4 = {} & A_{PPPP} + A_{NPPP} + A_{PNPP} + A_{NNPP} + A_{PPNP} + A_{NPNP} + A_{PNNP} + A_{NNNP} + A_{PPPN} + A_{NPPN} + A_{PNPN} \\ & + A_{NNPN} + A_{PPNN} + A_{NPNN} + A_{PNNN} + A_{NNNN} = \left(\left(r_3 \cdot R_{1PPP}\right) + \left(0 \cdot R_{2PPP}\right) + \left(r_6 \cdot R_{3PPP}\right)\right) + \left(\left(r_3 \cdot R_{1NPP}\right)\right. \\ & \left. + \left(0 \cdot R_{2NPP}\right) + \left(r_6 \cdot R_{3NPP}\right)\right) + \left(\left(r_3 \cdot R_{1PNP}\right) + \left(0 \cdot R_{2PNP}\right) + \left(r_6 \cdot R_{3PNP}\right)\right) + \left(\left(r_3 \cdot R_{1NNP}\right) + \left(0 \cdot R_{2NNP}\right)\right. \\ & \left. + \left(r_6 \cdot R_{3NNP}\right)\right) + \left(\left(r_3 \cdot R_{1PPN}\right) + \left(0 \cdot R_{2PPN}\right) + \left(r_6 \cdot R_{3PPN}\right)\right) + \left(\left(r_3 \cdot R_{1NPN}\right) + \left(0 \cdot R_{2NPN}\right) + \left(r_6 \cdot R_{3NPN}\right)\right) \\ & + \left(\left(r_3 \cdot R_{1PNN}\right) + \left(0 \cdot R_{2PNN}\right) + \left(r_6 \cdot R_{3PNN}\right)\right) + \left(\left(r_3 \cdot R_{1NNN}\right) + \left(0 \cdot R_{2NNN}\right) + \left(r_6 \cdot R_{3NNN}\right)\right) \end{aligned}$$

$$\begin{aligned} B_4 = {} & B_{PPPP} + B_{NPPP} + B_{PNPP} + B_{NNPP} + B_{PPNP} + B_{NPNP} \\ & + B_{PNNP} + B_{NNNP} + B_{PPPN} + B_{NPPN} + B_{PNPN} + B_{NNPN} \\ & + B_{PPNN} + B_{NPNN} + B_{PNNN} + B_{NNNN} = \left(\left(0 \cdot R_{1PPP}\right)\right. \\ & + \left(r_5 \cdot R_{2PPP}\right) + \left.\left(r_7 \cdot R_{3PPP}\right)\right) + \left(\left(0 \cdot R_{1NPP}\right) + \left(r_5 \cdot R_{2NPP}\right)\right. \\ & + \left.\left(r_7 \cdot R_{3NPP}\right)\right) + \left(\left(0 \cdot R_{1PNP}\right) + \left(r_5 \cdot R_{2PNP}\right)\right. \\ & + \left.\left(r_7 \cdot R_{3PNP}\right)\right) + \left(\left(0 \cdot R_{1NNP}\right) + \left(r_5 \cdot R_{2NNP}\right)\right. \\ & + \left.\left(r_7 \cdot R_{3NNP}\right)\right) + \left(\left(0 \cdot R_{1PPN}\right) + \left(r_5 \cdot R_{2PPN}\right)\right. \\ & + \left.\left(r_7 \cdot R_{3PPN}\right)\right) + \left(\left(0 \cdot R_{1NPN}\right) + \left(r_5 \cdot R_{2NPN}\right)\right. \\ & + \left.\left(r_7 \cdot R_{3NPN}\right)\right) + \left(\left(0 \cdot R_{1PNN}\right) + \left(r_5 \cdot R_{2PNN}\right)\right. \\ & + \left.\left(r_7 \cdot R_{3PNN}\right)\right) + \left(\left(0 \cdot R_{1NNN}\right) + \left(r_5 \cdot R_{2NNN}\right)\right. \\ & + \left.\left(r_7 \cdot R_{3NNN}\right)\right) \end{aligned}$$

$$\begin{aligned} C_4 = {} & C_{PPPP} + C_{NPPP} + C_{PNPP} + C_{NNPP} + C_{PPNP} + C_{NPNP} + C_{PNNP} \\ & + C_{NNNP} + C_{PPPN} + C_{NPPN} + C_{PNPN} + C_{NNPN} + C_{PPNN} \\ & + C_{NPNN} + C_{PNNN} + C_{NNNN} = \left(\left(0 \cdot R_{1PPP}\right) + \left(r_4 \cdot R_{2PPP}\right)\right. \\ & + \left.\left(0 \cdot R_{3PPP}\right)\right) + \left(\left(0 \cdot R_{1NPP}\right) + \left(r_4 \cdot R_{2NPP}\right) + \left(0 \cdot R_{3NPP}\right)\right) \\ & + \left(\left(0 \cdot R_{1PNP}\right) + \left(r_4 \cdot R_{2PNP}\right) + \left(0 \cdot R_{3PNP}\right)\right) + \left(\left(0 \cdot R_{1NNP}\right)\right. \\ & + \left(r_4 \cdot R_{2NNP}\right) + \left.\left(0 \cdot R_{3NNP}\right)\right) + \left(\left(0 \cdot R_{1PPN}\right) + \left(r_4 \cdot R_{2PPN}\right) + \left(0 \cdot R_{3PPN}\right)\right) \\ & + \left(\left(0 \cdot R_{1NPN}\right) + \left(r_4 \cdot R_{2NPN}\right) + \left(0 \cdot R_{3NPN}\right)\right) + \left(\left(0 \cdot R_{1PNN}\right)\right. \\ & + \left(r_4 \cdot R_{2PNN}\right) + \left.\left(0 \cdot R_{3PNN}\right)\right) + \left(\left(0 \cdot R_{1NNN}\right) + \left(r_4 \cdot R_{2NNN}\right) + \left(0 \cdot R_{3NNN}\right)\right) \end{aligned}$$

$$\begin{aligned}D_4 &= D_{PPPP} + D_{NPPP} + D_{PNPP} + D_{NNPP} + D_{PPNP} + D_{NPNP} + D_{PNNP}\\ &\quad + D_{NNNP} + D_{PPPN} + D_{NPPN} + D_{PNPN} + D_{NNPN} + D_{PPNN}\\ &\quad + D_{NPNN} + D_{PNNN} + D_{NNNN} = ((0 \cdot R_{1PPP}) + (0 \cdot R_{2PPP})\\ &\quad + (0 \cdot R_{3PPP})) + ((0 \cdot R_{1NPP}) + (0 \cdot R_{2NPP}) + (0 \cdot R_{3NPP}))\\ &\quad + ((0 \cdot R_{1PNP}) + (0 \cdot R_{2PNP}) + (0 \cdot R_{3PNP})) + ((0 \cdot R_{1NNP})\\ &\quad + (0 \cdot R_{2NNP}) + (0 \cdot R_{3NNP})) + ((0 \cdot R_{1PPN}) + (0 \cdot R_{2PPN})\\ &\quad + (0 \cdot R_{3PPN})) + ((0 \cdot R_{1NPN}) + (0 \cdot R_{2NPN}) + (0 \cdot R_{3NPN}))\\ &\quad + ((0 \cdot R_{1PNN}) + (0 \cdot R_{2PNN}) + (0 \cdot R_{3PNN})) + ((0 \cdot R_{1NNN})\\ &\quad + (0 \cdot R_{2NNN}) + (0 \cdot R_{3NNN}))\end{aligned}$$

$$\begin{aligned}E_4 &= E_{PPPP} + E_{NPPP} + E_{PNPP} + E_{NNPP} + E_{PPNP} + E_{NPNP} + E_{PNNP}\\ &\quad + E_{NNNP} + E_{PPPN} + E_{NPPN} + E_{PNPN} + E_{NNPN} + E_{PPNN} + E_{NPNN}\\ &\quad + E_{PNNN} + E_{NNNN} = ((0 \cdot R_{1PPP}) + (0 \cdot R_{2PPP}) + (0 \cdot R_{3PPP}))\\ &\quad + ((0 \cdot R_{1NPP}) + (0 \cdot R_{2NPP}) + (0 \cdot R_{3NPP})) + ((0 \cdot R_{1PNP})\\ &\quad + (0 \cdot R_{2PNP}) + (0 \cdot R_{3PNP})) + ((0 \cdot R_{1NNP}) + (0 \cdot R_{2NNP})\\ &\quad + (0 \cdot R_{3NNP})) + ((0 \cdot R_{1PPN}) + (0 \cdot R_{2PPN}) + (0 \cdot R_{3PPN}))\\ &\quad + ((0 \cdot R_{1NPN}) + (0 \cdot R_{2NPN}) + (0 \cdot R_{3NPN})) + ((0 \cdot R_{1PNN})\\ &\quad + (0 \cdot R_{2PNN}) + (0 \cdot R_{3PNN})) + ((0 \cdot R_{1NNN}) + (0 \cdot R_{2NNN}) + (0 \cdot R_{3NNN}))\end{aligned}$$

$$\begin{aligned}T_{e4} &= T_{ePPPP} + T_{eNPPP} + T_{ePNPP} + T_{eNNPP} + T_{ePPNP}\\ &\quad + T_{eNPNP} + T_{ePNNP} + T_{eNNNP} + T_{ePPPN} + T_{eNPPN} + T_{ePNPN} + T_{eNNPN}\\ &\quad + T_{ePPNN} + T_{eNPNN} + T_{ePNNN} + T_{eNNNN} = ((r_1 \cdot R_{1PPP})\\ &\quad + (r_2 \cdot R_{2PPP}) + (0 \cdot R_{3PPP})) + ((r_1 \cdot R_{1NPP}) + (r_2 \cdot R_{2NPP})\\ &\quad + (0 \cdot R_{3NPP})) + ((r_1 \cdot R_{1PNP}) + (r_2 \cdot R_{2PNP}) + (0 \cdot R_{3PNP}))\\ &\quad + ((r_1 \cdot R_{1NNP}) + (r_2 \cdot R_{2NNP}) + (0 \cdot R_{3NNP})) + ((r_1 \cdot R_{1PPN})\\ &\quad + (r_2 \cdot R_{2PPN}) + (0 \cdot R_{3PPN})) + ((r_1 \cdot R_{1NPN})\\ &\quad + (r_2 \cdot R_{2NPN}) + (0 \cdot R_{3NPN})) + ((r_1 \cdot R_{1PNN}) + (r_2 \cdot R_{2PNN})\\ &\quad + (0 \cdot R_{3PNN})) + ((r_1 \cdot R_{1NNN}) + (r_2 \cdot R_{2NNN}) + (0 \cdot R_{3NNN}))\end{aligned}$$

Expanding the brackets means:
Depth 4 predecessors:

$$\begin{aligned}A_4 &= r_3(R_{1PPP} + R_{1NPP} + R_{1PNP} + R_{1NNP} + R_{1PPN}\\ &\quad +R_{1NPN} + R_{1PNN} + R_{1NNN})\\ &\quad + 0(R_{2PPP} + R_{2NPP} + R_{2PNP} + R_{2NNP} + R_{2PPN}\\ &\quad +R_{2NPN} + R_{2PNN} + R_{2NNN}) + r_6(R_{3PPP}\\ &\quad +R_{3NPP} + R_{3PNP} + R_{3NNP} + R_{3PPN} + R_{3NPN}\\ &\quad +R_{3PNN} + R_{3NNN})\end{aligned}$$

$$\begin{aligned} B_4 = {} & 0\left(R_{1PPP} + R_{1NPP} + R_{1PNP} + R_{1NNP} + R_{1PPN} + R_{1NPN} + R_{1PNN} + R_{1NNN}\right) \\ & + r_5\left(R_{2PPP} + R_{2NPP} + R_{2PNP} + R_{2NNP} + R_{2PPN} + R_{2NPN} + R_{2PNN} + R_{2NNN}\right) \\ & + r_7\left(R_{3PPP} + R_{3NPP} + R_{3PNP} + R_{3NNP} + R_{3PPN} + R_{3NPN} + R_{3PNN} + R_{3NNN}\right) \end{aligned}$$

$$\begin{aligned} C_4 = {} & 0\left(R_{1PPP} + R_{1NPP} + R_{1PNP} + R_{1NNP} + R_{1PPN} + R_{1NPN} + R_{1PNN} + R_{1NNN}\right) \\ & + r_4\left(R_{2PPP} + R_{2NPP} + R_{2PNP} + R_{2NNP} + R_{2PPN} + R_{2NPN} + R_{2PNN} + R_{2NNN}\right) \\ & + 0\left(R_{3PPP} + R_{3NPP} + R_{3PNP} + R_{3NNP} + R_{3PPN} + R_{3NPN} + R_{3PNN} + R_{3NNN}\right) \end{aligned}$$

$$\begin{aligned} D_4 = {} & 0\left(R_{1PPP} + R_{1NPP} + R_{1PNP} + R_{1NNP} + R_{1PPN} + R_{1NPN} + R_{1PNN} + R_{1NNN}\right) \\ & + 0\left(R_{2PPP} + R_{2NPP} + R_{2PNP} + R_{2NNP} + R_{2PPN} + R_{2NPN} + R_{2PNN} + R_{2NNN}\right) \\ & + 0\left(R_{3PPP} + R_{3NPP} + R_{3PNP} + R_{3NNP} + R_{3PPN} + R_{3NPN} + R_{3PNN} + R_{3NNN}\right) \end{aligned}$$

$$\begin{aligned} E_4 = {} & 0\left(R_{1PPP} + R_{1NPP} + R_{1PNP} + R_{1NNP} + R_{1PPN} + R_{1NPN} + R_{1PNN} + R_{1NNN}\right) \\ & + 0\left(R_{2PPP} + R_{2NPP} + R_{2PNP} + R_{2NNP} + R_{2PPN} + R_{2NPN} + R_{2PNN} + R_{2NNN}\right) \\ & + 0\left(R_{3PPP} + R_{3NPP} + R_{3PNP} + R_{3NNP} + R_{3PPN} + R_{3NPN} + R_{3PNN} + R_{3NNN}\right) \end{aligned}$$

$$\begin{aligned} T_{e4} = {} & r_1\left(R_{1PPP} + R_{1NPP} + R_{1PNP} + R_{1NNP} + R_{1PPN} + R_{1NPN} + R_{1PNN} + R_{1NNN}\right) \\ & + r_2\left(R_{2PPP} + R_{2NPP} + R_{2PNP} + R_{2NNP} + R_{2PPN} + R_{2NPN} + R_{2PNN} + R_{2NNN}\right) \\ & + 0\left(R_{3PPP} + R_{3NPP} + R_{3PNP} + R_{3NNP} + R_{3PPN} + R_{3NPN} + R_{3PNN} + R_{3NNN}\right) \end{aligned}$$

$$A_4 = r_3\left(R_{13}\right) + 0\left(R_{23}\right) + r_6\left(R_{33}\right)$$

$$B_4 = 0\left(R_{13}\right) + r_5\left(R_{23}\right) + r_7\left(R_{33}\right)$$

$$C_4 = 0\left(R_{13}\right) + r_4\left(R_{23}\right) + 0\left(R_{33}\right)$$

$$D_4 = 0\left(R_{13}\right) + 0\left(R_{23}\right) + 0\left(R_{33}\right)$$

$$E_4 = 0\left(R_{13}\right) + 0\left(R_{23}\right) + 0\left(R_{33}\right)$$

$$T_{e4} = r_1\left(R_{13}\right) + r_2\left(R_{23}\right) + 0\left(R_{33}\right)$$

Depth 3 predecessors:

$$\begin{aligned} R_{13} & = R_{1PPP} + R_{1NPP} + R_{1PNP} + R_{1NNP} + R_{1PPN} + R_{1NPN} + R_{1PNN} + R_{1NNN} \\ & = \left(v_2 \cdot A_{PP}\right) + \left(v_2 \cdot A_{NP}\right) + \left(v_2 \cdot A_{PN}\right) + \left(v_2 \cdot A_{NN}\right) + \left(v_5 \cdot B_{PP}\right) + \left(v_5 \cdot B_{NP}\right) \\ & \quad + \left(v_5 \cdot B_{PN}\right) + \left(v_5 \cdot B_{NN}\right) + \left(v_1 \cdot C_{PP}\right) + \left(v_1 \cdot C_{NP}\right) + \left(v_1 \cdot C_{PN}\right) + \left(v_1 \cdot C_{NN}\right) \end{aligned}$$

$$\begin{aligned}R_{23} &= R_{2PPP} + R_{2NPP} + R_{2PNP} + R_{2NNP} + R_{2PPN} + R_{2NPN} + R_{2PNN} + R_{2NNN} \\ &= \left(v_7 \cdot B_{PP}\right) + \left(v_7 \cdot B_{NP}\right) + \left(v_7 \cdot B_{PN}\right) + \left(v_7 \cdot B_{NN}\right) + \left(v_4 \cdot C_{PP}\right) \\ &\quad + \left(v_4 \cdot C_{NP}\right) + \left(v_4 \cdot C_{PN}\right) + \left(v_4 \cdot C_{NN}\right) + \left(v_6 \cdot E_{PP}\right) \\ &\quad + \left(v_6 \cdot E_{NP}\right) + \left(v_6 \cdot E_{PN}\right) + \left(v_6 \cdot E_{NN}\right)\end{aligned}$$

$$\begin{aligned}R_{33} &= R_{3PPP} + R_{3NPP} + R_{3PNP} + R_{3NNP} + R_{3PPN} + R_{3NPN} + R_{3PNN} + R_{3NNN} \\ &= \left(v_8 \cdot A_{PP}\right) + \left(v_8 \cdot A_{NP}\right) + \left(v_8 \cdot A_{PN}\right) + \left(v_8 \cdot A_{NN}\right) + \left(v_9 \cdot B_{PP}\right) + \left(v_9 \cdot B_{NP}\right) \\ &\quad + \left(v_9 \cdot B_{PN}\right) + \left(v_9 \cdot B_{NN}\right) + \left(v_{10} \cdot D_{PP}\right) + \left(v_{10} \cdot D_{NP}\right) + \left(v_{10} \cdot D_{PN}\right) \\ &\quad + \left(v_{10} \cdot D_{NN}\right) + \left(v_3 \cdot A_{PP}\right) + \left(v_3 \cdot A_{NP}\right) + \left(v_3 \cdot A_{PN}\right) + \left(v_3 \cdot A_{NN}\right)\end{aligned}$$

$$R_{13} = v_2\left(A_{PP} + A_{NP} + A_{PN} + A_{NN}\right) + v_5\left(B_{PP} + B_{NP} + B_{PN} + B_{NN}\right) + v_1\left(C_{PP} + C_{NP} + C_{PN} + C_{NN}\right)$$

$$R_{23} = v_7\left(B_{PP} + B_{NP} + B_{PN} + B_{NN}\right) + v_4\left(C_{PP} + C_{NP} + C_{PN} + C_{NN}\right) + v_6\left(E_{PP} + E_{NP} + E_{PN} + E_{NN}\right)$$

$$\begin{aligned}R_{33} &= v_8\left(A_{PP} + A_{NP} + A_{PN} + A_{NN}\right) + v_9\left(B_{PP} + B_{NP} + B_{PN} + B_{NN}\right) \\ &\quad + v_{10}\left(D_{PP} + D_{NP} + D_{PN} + D_{NN}\right) + v_3\left(A_{PP} + A_{NP} + A_{PN} + A_{NN}\right)\end{aligned}$$

$$R_{13} = v_2\left(A_2\right) + v_5\left(B_2\right) + v_1\left(C_2\right)$$

$$R_{23} = v_7\left(B_2\right) + v_4\left(C_2\right) + v_6\left(E_2\right)$$

$$R_{33} = v_3\left(A_2\right) + v_8\left(A_2\right) + v_9\left(B_2\right) + v_{10}\left(D_2\right)$$

Depth 2 predecessors:

$$\begin{aligned}A_2 &= A_{PP} + A_{NP} + A_{PN} + A_{NN} = (\left(r_3 \cdot R_{1P}\right) + \left(0 \cdot R_{2P}\right) + \left(r_6 \cdot R_{3P}\right)) + (\left(r_3 \cdot R_{1P}\right) \\ &\quad + \left(0 \cdot R_{2P}\right) + \left(r_6 \cdot R_{3P}\right)) + (\left(r_3 \cdot R_{1N}\right) + \left(0 \cdot R_{2N}\right) + \left(r_6 \cdot R_{3N}\right)) + (\left(r_3 \cdot R_{1N}\right) \\ &\quad + \left(0 \cdot R_{2N}\right) + \left(r_6 \cdot R_{3N}\right))\end{aligned}$$

$$\begin{aligned}B_2 &= B_{PP} + B_{NP} + B_{PN} + B_{NN} = (\left(0 \cdot R_{1P}\right) + \left(r_5 \cdot R_{2P}\right) + \left(r_7 \cdot R_{3P}\right)) + (\left(0 \cdot R_{1P}\right) \\ &\quad + \left(r_5 \cdot R_{2P}\right) + \left(r_7 \cdot R_{3P}\right)) + (\left(0 \cdot R_{1N}\right) + \left(r_5 \cdot R_{2N}\right) + \left(r_7 \cdot R_{3N}\right)) + (\left(0 \cdot R_{1N}\right) \\ &\quad + \left(r_5 \cdot R_{2N}\right) + \left(r_7 \cdot R_{3N}\right))\end{aligned}$$

$$\begin{aligned}C_2 &= C_{PP} + C_{NP} + C_{PN} + C_{NN} = (\left(0 \cdot R_{1P}\right) + \left(r_4 \cdot R_{2P}\right) + \left(0 \cdot R_{3P}\right)) + (\left(0 \cdot R_{1P}\right) \\ &\quad + \left(r_4 \cdot R_{2P}\right) + \left(0 \cdot R_{3P}\right)) + (\left(0 \cdot R_{1N}\right) + \left(r_4 \cdot R_{2N}\right) + \left(0 \cdot R_{3N}\right)) + (\left(0 \cdot R_{1N}\right) \\ &\quad + \left(r_4 \cdot R_{2N}\right) + \left(0 \cdot R_{3N}\right))\end{aligned}$$

$$D_2 = D_{PP} + D_{NP} + D_{PN} + D_{NN} = ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N}))$$

$$E_2 = E_{PP} + E_{NP} + E_{PN} + E_{NN} = ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N}))$$

$$T_{e2} = T_{ePP} + T_{eNP} + T_{ePN} + T_{eNN} = ((r_1 \cdot R_{1P}) + (r_2 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((r_1 \cdot R_{1P}) + (r_2 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((r_1 \cdot R_{1N}) + (r_2 \cdot R_{2N}) + (0 \cdot R_{3N})) + ((r_1 \cdot R_{1N}) + (r_2 \cdot R_{2N}) + (0 \cdot R_{3N}))$$

The relationships in the depth 1 predecessor stage (succeeding the variables in the depth 2 predecessor stage) are either positive or negative, therefore either $A_{PP} + A_{NP}$ or $A_{PN} + A_{NN}$ will be zero, and hence the value of the variables in the depth 2 predecessor stage will be:

Depth 2 predecessors:

$$A_2 = A_{PP} + A_{NP} + A_{PN} + A_{NN} = ((r_3 \cdot R_{1P}) + (0 \cdot R_{2P}) + (r_6 \cdot R_{3P})) + ((r_3 \cdot R_{1N}) + (0 \cdot R_{2N}) + (r_6 \cdot R_{3N})) + 0 + 0$$

$$B_2 = B_{PP} + B_{NP} + B_{PN} + B_{NN} = ((0 \cdot R_{1P}) + (r_5 \cdot R_{2P}) + (r_7 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (r_5 \cdot R_{2N}) + (r_7 \cdot R_{3N})) + 0 + 0$$

$$C_2 = C_{PP} + C_{NP} + C_{PN} + C_{NN} = ((0 \cdot R_{1P}) + (r_4 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (r_4 \cdot R_{2N}) + (0 \cdot R_{3N})) + 0 + 0$$

$$D_2 = D_{PP} + D_{NP} + D_{PN} + D_{NN} = ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N})) + 0 + 0$$

$$E_2 = E_{PP} + E_{NP} + E_{PN} + E_{NN} = ((0 \cdot R_{1P}) + (0 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((0 \cdot R_{1N}) + (0 \cdot R_{2N}) + (0 \cdot R_{3N})) + 0 + 0$$

$$T_{e2} = T_{ePP} + T_{eNP} + T_{ePN} + T_{eNN} = ((r_1 \cdot R_{1P}) + (r_2 \cdot R_{2P}) + (0 \cdot R_{3P})) + ((r_1 \cdot R_{1N}) + (r_2 \cdot R_{2N}) + (0 \cdot R_{3N})) + 0 + 0$$

$$A_2 = 0(R_{2P} + R_{2N}) + r_6(R_{3P} + R_{3N}) + r_3(R_{1P} + R_{1N})$$

$$B_2 = 0(R_{1P} + R_{1N}) + r_5(R_{2P} + R_{2N}) + r_7(R_{3P} + R_{3N})$$

$$C_2 = 0\left(R_{1P} + R_{1N}\right) + r_4\left(R_{2P} + R_{2N}\right) + 0\left(R_{3P} + R_{3N}\right)$$

$$D_2 = 0\left(R_{1P} + R_{1N}\right) + 0\left(R_{2P} + R_{2N}\right) + 0\left(R_{3P} + R_{3N}\right)$$

$$E_2 = 0\left(R_{1P} + R_{1N}\right) + 0\left(R_{2P} + R_{2N}\right) + 0\left(R_{3P} + R_{3N}\right)$$

$$T_{e2} = r_1\left(R_{1P} + R_{1N}\right) + r_2\left(R_{2P} + R_{2N}\right) + 0\left(R_{3P} + R_{3N}\right)$$

$$A_2 = 0\left(R_{21}\right) + r_6\left(R_{31}\right) + r_3\left(R_{11}\right)$$

$$B_2 = 0\left(R_{11}\right) + r_5\left(R_{21}\right) + r_7\left(R_{31}\right)$$

$$C_2 = 0\left(R_{11}\right) + r_4\left(R_{21}\right) + 0\left(R_{31}\right)$$

$$D_2 = 0\left(R_{11}\right) + 0\left(R_{21}\right) + 0\left(R_{31}\right)$$

$$E_2 = 0\left(R_{11}\right) + 0\left(R_{21}\right) + 0\left(R_{31}\right)$$

$$T_{e2} = r_1\left(R_{11}\right) + r_2\left(R_{21}\right) + 0\left(R_{31}\right)$$

Depth 1 predecessors:

$$R_{11} = R_{1P} + R_{1N} = \left(v_5 \cdot B_0\right) + \left(v_1 \cdot C_0\right) + \left(v_2 \cdot A_0\right)$$

$$R_{21} = R_{2P} + R_{2N} = \left(v_6 \cdot E_0\right) + \left(v_7 \cdot B_0\right) + \left(v_4 \cdot C_0\right)$$

$$R_{31} = R_{3P} + R_{3N} = \left(v_3 \cdot A_0\right) + \left(v_{10} \cdot D_0\right) + \left(v_8 \cdot A_0\right) + \left(v_9 \cdot B_0\right)$$

$$R_{11} = \left(v_5 \cdot B_0\right) + \left(v_1 \cdot C_0\right) + \left(v_2 \cdot A_0\right)$$

$$R_{21} = \left(v_6 \cdot E_0\right) + \left(v_7 \cdot B_0\right) + \left(v_4 \cdot C_0\right)$$

$$R_{31} = \left(v_3 \cdot A_0\right) + \left(v_{10} \cdot D_0\right) + \left(v_8 \cdot A_0\right) + \left(v_9 \cdot B_0\right)$$

Calculating all phases then simplifies to:
Target array:

$$A_0$$

$$B_0$$

$$C_0$$

$$D_0$$

$$E_0$$

$$T_{e0}$$

Depth 1 predecessors:

$$R_{11} = v_2(A_0) + v_5(B_0) + v_1(C_0)$$

$$R_{21} = v_7(B_0) + v_4(C_0) + v_6(E_0)$$

$$R_{31} = v_3(A_0) + v_8(A_0) + v_9(B_0) + v_{10}(D_0)$$

Depth 2 predecessors:

$$A_2 = 0(R_{21}) + r_6(R_{31}) + r_3(R_{11})$$

$$B_2 = 0(R_{11}) + r_5(R_{21}) + r_7(R_{31})$$

$$C_2 = 0(R_{11}) + r_4(R_{21}) + 0(R_{31})$$

$$D_2 = 0(R_{11}) + 0(R_{21}) + 0(R_{31})$$

$$E_2 = 0(R_{11}) + 0(R_{21}) + 0(R_{31})$$

$$T_{e2} = r_1(R_{11}) + r_2(R_{21}) + 0(R_{31})$$

Depth 3 predecessors:

$$R_{13} = v_2(A_2) + v_5(B_2) + v_1(C_2)$$

$$R_{23} = v_7(B_2) + v_4(C_2) + v_6(E_2)$$

$$R_{33} = v_3(A_2) + v_8(A_2) + v_9(B_2) + v_{10}(D_2)$$

And the Final rank set would then be defined as follows:
Depth 4 predecessors:

$$A_4 = r_3(R_{13}) + 0(R_{23}) + r_6(R_{33})$$

$$B_4 = 0(R_{13}) + r_5(R_{23}) + r_7(R_{33})$$

$$C_4 = 0(R_{13}) + r_4(R_{23}) + 0(R_{33})$$

$$D_4 = 0(R_{13}) + 0(R_{23}) + 0(R_{33})$$

$$E_4 = 0(R_{13}) + 0(R_{23}) + 0(R_{33})$$

$$T_{e4} = r_1(R_{13}) + r_2(R_{23}) + 0(R_{33})$$

This is equivalent to a series of matrix operations, which can be performed on the adjacency matrix of the graph, the values of all nodes in each set being calculated as follows:

Adjacency matrix:

$$\begin{bmatrix} \ddots & A & B & C & D & E & T_e & R_1 & R_2 & R_3 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & r_3 & r_6 & 0 \\ B & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_5 & r_7 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_4 & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 0 & r_1 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 & 0 \\ T_e & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_1 & v_2 & v_5 & v_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_2 & 0 & v_7 & v_4 & 0 & v_6 & 0 & 0 & 0 & 0 \\ R_3 & v_8 + v_3 & v_9 & 0 & v_{10} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

To get depth 1 predecessors:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & r_3 & r_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_5 & r_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & v_5 & v_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & v_7 & v_4 & 0 & v_6 & 0 & 0 & 0 & 0 \\ v_8 + v_3 & v_9 & 0 & v_{10} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_0 \\ B_0 \\ C_0 \\ D_0 \\ E_0 \\ T_{e,0} \\ R_{1,0} \\ R_{2,0} \\ R_{3,0} \end{bmatrix} = \begin{bmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \\ E_1 \\ T_{e,1} \\ R_{1,1} \\ R_{2,1} \\ R_{3,1} \end{bmatrix}$$

To get depth 2 predecessors:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & r_3 & r_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_5 & r_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & v_5 & v_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & v_7 & v_4 & 0 & v_6 & 0 & 0 & 0 & 0 \\ v_8 + v_3 & v_9 & 0 & v_{10} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \\ E_1 \\ T_{e,1} \\ R_{1,1} \\ R_{2,1} \\ R_{3,1} \end{bmatrix} = \begin{bmatrix} A_2 \\ B_2 \\ C_2 \\ D_2 \\ E_2 \\ T_{e,2} \\ R_{1,2} \\ R_{2,2} \\ R_{3,2} \end{bmatrix}$$

To get depth 3 predecessors:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & r_3 & r_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_5 & r_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & v_5 & v_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & v_7 & v_4 & 0 & v_6 & 0 & 0 & 0 & 0 \\ v_8 + v_3 & v_9 & 0 & v_{10} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_2 \\ B_2 \\ C_2 \\ D_2 \\ E_2 \\ T_{e,2} \\ R_{1,2} \\ R_{2,2} \\ R_{3,2} \end{bmatrix} = \begin{bmatrix} A_3 \\ B_3 \\ C_3 \\ D_3 \\ E_3 \\ T_{e,3} \\ R_{1,3} \\ R_{2,3} \\ R_{3,3} \end{bmatrix}$$

To get depth 4 predecessors:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & r_3 & r_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_5 & r_7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_2 & v_5 & v_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & v_7 & v_4 & 0 & v_6 & 0 & 0 & 0 & 0 \\ v_8 + v_3 & v_9 & 0 & v_{10} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_3 \\ B_3 \\ C_3 \\ D_3 \\ E_3 \\ T_{e,3} \\ R_{1,3} \\ R_{2,3} \\ R_{3,3} \end{bmatrix} = \begin{bmatrix} A_4 \\ B_4 \\ C_4 \\ D_4 \\ E_4 \\ T_{e,4} \\ R_{1,4} \\ R_{2,4} \\ R_{3,4} \end{bmatrix}$$

The above matrix operations can be represented in the following notation:

$$x_i^{(n+1)} = \sum a_{i,j} x_i^{(n)}$$

where x_i is the *i*th column of the node ranking vector, $a_{i,j}$ is the *j*th row of the *i*th column, and n is the depth of predecessor from the target node. The initial node ranking vector (n = 0) with the index of each node set to 0, except the target node which is set to 1.

All the above is effectively a simplified generalisation of the OCARINA (Optimal Condition Approaching via Reaction-In-Network Analysis) algorithm in [15] using matrix operations. Taking account of the generalisation into a Variable-Relationship graph, the same acronym can still be used, with a slight modification: Optimal Condition Approaching via *Relationship*-In-Network Analysis.

References

1. Bruggeman PJ, Iza F, Brandenburg R (2017) Foundations of atmospheric pressure non-equilibrium plasmas. Plasma Sources Sci Technol 26:12
2. Doran PM (2013) Chapter 14 - Reactor Engineering". In: Doran PM (ed) Bioprocess Engineering Principles (Second Edition). Academic press, pp 761–852
3. Levenspiel O (1972) Chemical Reaction Engineering. Wiley, New York
4. Harary F, Norman RZ, Cartwright D (1965) Structural models An introduction to the theory of directed graphs. Wiley
5. Anderson JL, Flamm C, Merkle D, Stadler PF (2017) An intermediate level of abstraction for computational systems chemistry. Philos Trans R Soc A 375:20160354
6. Fagerberg R, Flamm C, Kianian R, Merkle D, Stadler PF (2018) Finding the K best synthesis plans. J Cheminform. https://doi.org/10.1186/s13321-018-0273-z
7. H. G. Othmer, 1981 "A GRAPH-THEORETIC ANALYSIS OF CHEMICAL REACTION NETWORKS I . Invariants , Network Equivalence and Nonexistence of Various Types of Steady States," *Lect. Notes*, pp. 1–36
8. Andersen JL, Merkle D (2018) A Generic Framework for Engineering Graph Canonization Algorithms. SIAM, Philadelphia, PA, pp 139–153
9. Mondal S, Mandal B (2021) Reaction kinetic graphs of chain reactions: Solutions for their rate equations. Chem Phys Lett 781:138977

10 McDermott MJ, Dwaraknath SS, Persson KA (2021) A graph-based network for predicting chemical reaction pathways in solid-state materials synthesis. Nat Commun. https://doi.org/10.1038/s41467-021-23339-x

11. Sakamoto A, Kawakami H, Yoshikawa K (1988) A graph theoretical approach to complex reaction networks. Chem Phys Lett 146(5):444–448
12. Craciun G, Feinberg M (2006) Multiple equilibria in complex chemical reaction networks: II. The species-reaction graph. SIAM J Appl Math 66(4):1321–1338
13. Domijan M, Kirkilionis M (2008) Graph theory and qualitative analysis of reaction networks. Netw Heterog Med 3(2):295–322

14 Silva SJ, Burrows SM, Evans MJ, Halappanavar M (2021) A graph theoretical intercomparison of atmospheric chemical mechanisms. Geophys Res Lett. https://doi.org/10.1029/2020GL090481

15. Holmes TD, Rothman RH, Zimmerman WB (2021) Graph theory applied to plasma chemical reaction engineering. Plasma Chem Plasma Process 41(2):531–557
16. J. L. Anderson, C. Flamm, D. Merkle, P. F. Stadler, J. L. Andersen, and C. Flamm, "Chemical Transformation Motifs -Modelling Pathways as Integer Hyperflows." 2016.

17 Murakami T, Sakai O (2020) Rescaling the complex network of low-temperature plasma chemistry through graph-theoretical analysis. Plasma Sources Sci Technol 29:115018

18. Lehman R (2004) An algorithm for the determination of all significant pathways in chemical reaction systems. J Atmos Chem 47:45–78
19. Markosyan AH, Luque A, Gordillo-Vazquez FJ, Ebert U (2014) PumpKin: a tool to find principal pathways in plasma chemical models. Comput Phys Commun 185:2697–2702
20. TadayonMousavi S, Gulpen JGM, Graef WAAD, Koelman PMJ, Carbone EAD, van Dijk J (2022) Assessment of the suitability of the chemical reaction pathway algorithm as a reduction method for plasma chemistry. J Phys D Appl Phys 55:505201
21. M. J. et al. M. Bastian, S. Heymann, "Gephi: an open source software for exploring and manipulating networks," in *Proceedings of International AAAI Conference on Web and Social Media*, 2009.

22 Andersen JL, Flamm C, Merkle D, Stadler PF (2012) Maximizing output and recognizing autocatalysis in chemical reaction networks is NP-complete. J Syst Chem. https://doi.org/10.1186/1759-2208-3-1

23. Bonacich P (2007) Some unique properties of eigenvector centrality. Soc Networks 29(4):555–564
24. Katz L (1953) A new status index derived from sociometric analysis. Psychmetrika 18(1):39–43
25. Newman MEJ (2010) Networks: An Introduction. Oxford University Press, Oxford
26. P. E. Black, "'depth,'" *Dictionary of Algorithms and Data Structures [online]*, 2011. [Online]. Available: https://www.nist.gov/dads/HTML/depth.html. [Accessed: 26-May-2011].
27. Sakiyama Y, Graves DB, Chang H-W, Shimizu T, Morfill GE (2012) Plasma chemistry model of surface microdischarge in humid air and dynamics of reactive neutral species. J Phys D Appl Phys 45(42):425201
28. Eliasson B, Kogelschatz U (1991) Modeling and applications of silent discharge plasmas. IEEE Trans Plasma Sci 19(2):309–323
29. A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," *7th Python in Science Conference (SciPy 2008)*. pp. 11–15, 2008.

30. Meurer A et al (2017) SymPy: symbolic computing in python. PeerJ Comput Sci 1:2017
31. W. B. J. Zimmerman, *Process Modelling and Simulation with Finite Element Methods*. 2004.
32. S. P. Borgatti, M. G. Everett, and J. C. Johnson, *Analyzing Social Networks*. SAGE Publications Ltd, 2013.
33. Lawyer G (2015) Understanding the influence of all nodes in a network. Sci Rep 5:1–9

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.