17th European Symposium on Computer Aided Process Engineering – ESCAPE17
V. Plesu and P.S. Agachi (Editors)

A Graph-Theory-Base Approach to the Analysis of Large-Scale Plants

Heinz A Preisig

Department of Chemical Engineering NTNU, Trondheim, Norway
e-mail: Heinz.Preisig@chemeng.ntnu.no

Abstract

On-line balancing of mass and energy in a large-scale plant is a feasible operation given the development state of the current data-acquisition systems. However, not all data are available in real time and the programmed version of the instrumentation flow sheets in form of a graph cannot be used directly. They need to be modified so as to match the available information.

Two cases are discussed: the dynamic case, where all units are seen as dynamic components and the steady state case, where each unit is assumed to operate at steady state. The analysis is done purely on a graph basis. The idea is the essential part; the resulting algorithms are extremely simple and only require a path search algorithm such as depth-first or a breath-first search algorithm.

Keywords: Real-time data acquisition, graph analysis, flow sheeting

1. Introduction

Today's large-scale plants, such as refineries, are equipped with plant-wide data acquisition equipment that log a large amount of process data as part of the plant's operation. This data is primarily used for planning but may also be used for plant-wide control scheduling of maintenance, fault detection and simply to keep a statistics of the operations.

We are interested in assessing the plant's operations in terms of mass and energy of parts of the plant, preferably to the detail possible. This requires closing the mass and energy balances over parts of the plant. In a large plant, where one has hundreds of streams and correspondingly many measurements, it is not trivial to identify the parts for which balances can be drawn up thereby also identifying what type of quantity can be balanced in the respective case. Thus our first effort was to work on a method to systematically determine on what balances we can actually draw up in the plant.

2. How the Model is Constructed

This group is dedicated to network modelling whereby the plant is represented as a network of primitive capacities that communicate mass and energy [3]. We first map the physical containment of the plant into a network of control volumes and communications of material and energy between them. Each control volume represents a capacity for mass and energy and each connection represents a flow of mass or energy. The result is a directed graph with the nodes representing the different control volumes and the edges representing the connections between connected pairs of control volumes. Arcs can thus not split or join as this would be the case in a flow sheet. Both, the vertices, as well as the arcs can be typed. For example one distinguishes between mass transfer and conductive heat transfer as well as mechanical work.

In the second step, (chemical) species are being introduced by “injecting” them at their respective source. Such sources are often modelled as infinite large capacities. The graph is analysed for its connectivity[1]. If one assigns to every type of transferred and conserved quantity a colour and defines a set of colour-coordination rules, which reflect component mass and energy balances, it is easy to construct the domains in which for example species exist (= species domain), or mass is exchanged (= mass domain) or heat is exchanged (= “heat” domain). Colour combinations, defined by the rules, allow also the computation of where what balance can be drawn.

The question we asked was what we can compute given the graph and the online data, which are available for some of the streams. In our case we have always too little information. Thus it is a priori clear that the balances over the different control volumes cannot be computed. However, we allow for reconfiguration of the graph in the sense that we allow for agglomeration by simply adding groups of connected control volumes together. Adding a set of connected control volumes will represent the corresponding total volume of the group with the internal boundaries being eliminated. In terms of the graph, the arcs inside the agglomerated part are eliminated. In terms of the conservation principles applied to the extensive quantities in each node (control volume) the agglomeration operation eliminates all internal flows.

[1] Notation see Wikepedia : http://en.wikipedia.org/wiki/Glossary_of_graph_theory

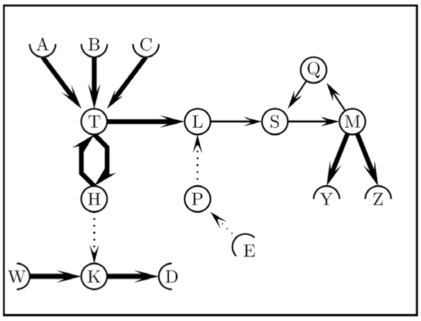
The question is then refined to: what sub-graphs need to be agglomerated in order to compute what-ever-else can be computed in the graph?

We shall next discuss the dynamic and the static graph. In order to simplify the discussion, we shall not type the extensive quantity but discuss the case of one only, say mass, to mention a common case. We shall also assume that no reaction occurs. The extension to other conserved quantities is straightforward.

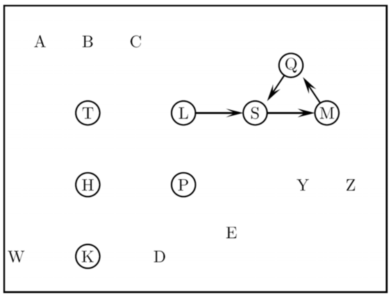
3. Developing the Algorithms

First we look at the dynamic case. Here each node represents a dynamic system. We ask the question what parts of the graph representing the plant with a certain granularity need to be agglomerated given a set of connections marked as being observed (full line) with the rest not being observed (thin line) by the appropriate measurement equipment. Whilst the problem can be wrapped into mathematics, a pure graphical explanation has been chosen.

3.1. Dynamic Case



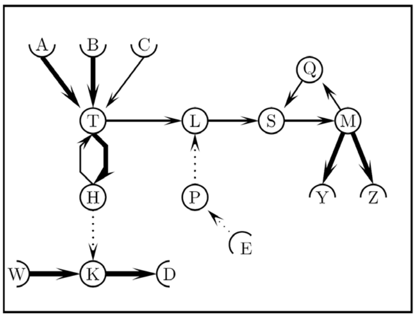
The base case has two types of connections, shown as dotted and full lines. First we eliminate the dotted ones, which could represent heat streams for example, as we only analyse the monochromatic case. By definition, the reservoirs, being infinitely large systems shown as open half circles, cannot be balanced and must be removed as well.



In order to compute a dynamic node without internal dynamic such as reactions, one requires knowledge of ALL connections crossing the node boundary. It is apparent that one can calculate node T, H and K, whilst L, S, M and Q are not completely defined. Thus: which of the nodes must be agglomerated in order to compute the resulting agglomerated graph? In the example it is easy to see what the solution is, namely the nodes L, S, M and Q must be agglomerated for the sum of it is fully determined.

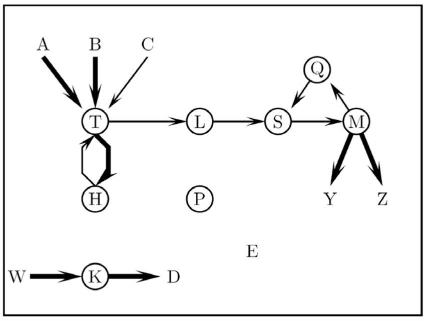
The algorithmic solution is astonishingly simple: if one deletes all known streams and ignores the reservoirs, one is left with the sub-graphs that need to be agglomerated. Thus all one requires is a graph algorithm that determines the components of the graph, the connected sub-graphs, which are easily available or programmed as essentially it is only a depth first search algorithm.

3.2. Steady-State Case



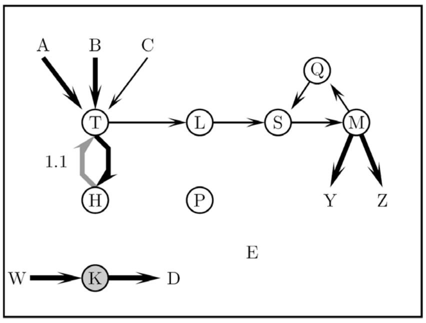
It turned out that this is a much harder problem. Many different approaches were tried. Complex methods such as he celebrated Dulmage Mendelsohn decomposition, which did yield results for bipartite graphs only; but once the idea has come up, the resulting algorithm got stunningly simple:

To demonstrate the problem, we take the topology of the previous example, but define a slightly different set of known streams.



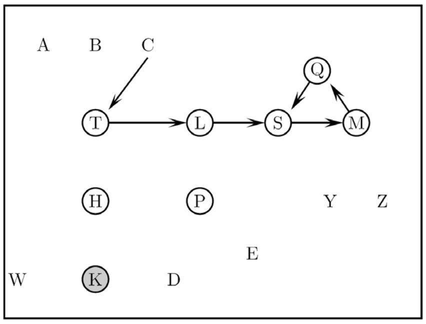
Again, we work with the monochromatic case thus remove the dotted lines focusing on the full lines representing mass flow only. Also balancing reservoirs is without meaning and consequently the reservoirs are removed from the graph.

We can now compute the stream H → T. We mark it with a grey thick arrow and set it first into the first list of computations (1.1) then search further if we can identify a node with the same structure.



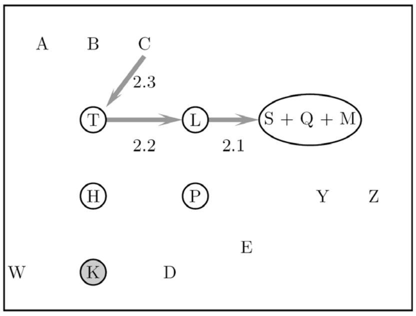
This is not the case. Though we did also find a node for which all the streams are known, namely node K. This node is over-determined (see below) and is marked with a grey background.

So in the next stage, we delete all known and computable connections and are left with a set of graph components including T, L, M, S and Q. All other components of the graph are single nodes. The only way to compute any other connection is to try and agglomerate a group of nodes. Since we seek nodes with one unknown connection, we need to define the boundary of the agglomerated graph



such that there is one unknown connection only. Cutting through any loop does leave us with two unknown streams. Thus we must eliminate the circuits by agglomerating the nodes forming a circuit, here S, M and Q.

Now we repeat searching for nodes that have one unknown connection only and find as designed node (S+Q+M). After marking the computable connection L → (S+Q+M) we repeat the search and find next the connection T → L and C → T. Again we put them in the given sequence in the list of computable connections. Deleting the computable connections leaves us with primitive graph components only and the algorithm terminates. In other cases one would again seek a loop in a graph component and repeat the last step of agglomeration followed by a search for nodes with one unknown connection until only primitive graph components are left.



The result of the algorithm is

- A list of nodes representing reservoirs: A, B, C, W, D, E, Y Z.
- A sequences of triple of lists (agglomeration of nodes, computable nodes and connections, over-determined nodes):
  - step 1: (agglomeration : -; computable: H → T from H; over-det.: K)
  - step 2: (agglomeration: S, Q, M, computable: L → (S+Q+M) from (S+Q+M), T → L from L, C → T from T; over-det: -).

Extension: The discussion was limited to the mono-chromatic case, meaning for one type of connection only. The extension to other cases is trivial: For mass only, one only requires the total mass, which may also be obtained from summing all component masses in a stream. To obtain the component information, all components in the respective streams must be known and finally to compute energy streams, one need to know the mass streams and its thermodynamic properties, all heat streams and all the mechanical work streams.

4. Implementation

Given a graph representation of a plant and given a set of measurements, the analysis can be done as described and the result can be implemented into the tool that analyses the plant on-line. The task can be synchronised with the sampling of the measurements in the plant taking the information from the real-time data base through an appropriate interface protocol.

Over-determined can be seen as dynamic nodes and may give information on how much material is accumulated in the part of the plant represented by the node. Thus one can for example observe the accumulation in a tank etc. Overdetermined nodes may also result from the agglomeration process.

In the case of assuming steady state for the different nodes, a circuit finding algorithm is required as well. The latter can though be constructed from a deptfirst algorithm in two stages: move first from a start node to an adjacent node, then break the connection behind and find a path from the adjacent node to the starting node. If it exists a circuit has been identified.

5. Conclusions

The problem was formulated on what can be calculated when incomplete stream information is available from the plant. A literature search revealed no results that address this problem specifically though the use of graph theoretical approaches in the context of representing and solving models describing the operation of chemical plants is an old and correspondingly rich field [1,2]. So to author's knowledge the problem formulation is new and correspondingly the presented results. Both the dynamic case and the steady state case are being analysed.
Both algorithms are surprisingly simple and require only a path-search algorithm for example a depth-first search algorithm.The algorithms provide agglomeration sets of nodes and computation sequences.Acknowledgements

The author thanks Alice Olsen and Ida Julseth Gjerde for their work on the project and Statoil for providing the problem and funding the activity.

References

1. Freidler F., Tarjan K., Huang Y. W., Fan L. T; Graph-theoretic approach to process synthesis axioms and theorems; Chem Eng Sci, vol 47, no 8, (1992), pp1973-1988
2. Mah, Richard S. H; Application of graph theory to process design and analysis; Comp Chem Eng, vol 7, (1983), p 239-257
3. Westerweele, M R; Five steps for building consistent dynamic process models and their implementation in the computer tool MODELLER; PhD Thesis, Eindhoven Uni of Tech, Netherlands, 2003