CHEMICAL ENGINEERING TRANSACTIONS Volume 21, 2010
Editor J. J. Klemeš, H. L. Lam, P. S. Varbanov
Copyright © 2010, AIDIC Servizi S.r.l., ISBN 978-88-95608-05-1 ISSN 1974-9791
DOI: 10.3303/CET1021232

PNS Solutions: a P-Graph Based Programming Framework for Process Network Synthesis

Virag Varga[1], Istvan Heckl[1], Ferenc Friedler[1*], L. T. Fan[2]
[1]Department of Computer Science and Systems Technology, University of Pannonia
Egyetem u. 10, 8200 Veszprém, Hungary,
friedler@dcs.uni-pannon.hu
[2]Department of Chemical Engineering, Kansas State University, USA

A novel programming framework has been developed for the P-graph-based methodology to provide a standardized software environment for different classes of process-network synthesis (PNS) problems. The P-graph framework has been proven highly successful; its applicability encompasses wide-ranging areas such as reaction-pathway identification, vehicle-routing problems and business-process modeling. A uniform programming paradigm has been proposed here to integrate various available solution engines and interfaces of different types of PNS problems into a single system. A client server architecture with a standardized communication protocol has also been developed, which renders it to be deployable by various client programs with different features, possibly implementable in different programming languages, or adoptable by varied solvers customized for specific problems. Numerous P-graph-based algorithms have been implemented to demonstrate the efficacy of the P-graph framework.

1. Introduction

A process network creates the desired products from specific raw materials with a given set of operating units. An operating unit can be any entity that transforms given inlet materials into given outlet materials, e.g., a reactor and distillation column. The ratios of the inlet and outlet materials are parts of the problem definition. The objective of process-network synthesis (PNS) is to identify the most favorable, i.e., optimum, process-network (Biegler et al., 1997; Pahor et al., 2001; Westerberg, 2004).
The P-graph-based methodology (Friedler et al., 1992; 1993) has given rise to a graph theoretical approach for solving PNS problems. P-graphs (process graphs) are bipartite graphs, each comprising nodes for a set of materials, a set of operating units, and arcs linking them. The materials can be the raw materials, intermediates, or products. The operating units are defined in terms of input and output materials as well as their ratios.
A solution structure of a PNS problem is deemed combinatorially feasible if it satisfies the following five axioms (Friedler et al., 1993): (i) every demand is represented in the structure; (ii) a material represented in the structure is a resource if and only if it is not an output from any operating unit represented in the structure; (iii) every operating unit represented in the structure is defined in the synthesis problem; (iv) any operating unit

represented in the structure has at least one directed path leading to a product; and (v) if a material belongs to the structure, it must be an input to or output from at least one operating unit represented in the structure.

The union of all the solution structures is termed the maximal structure. An effective algorithm, algorithm MSG, has been developed for generating the maximal structure in polynomial time. The maximal structure is also the initial point of finding all the solution structures and the optimal structure.

The solution structure generator (algorithm SSG) determines exhaustively the combinatorally feasible solution structures capable of producing every desired product (Friedler et al, 1995). The input to algorithm SSG includes the products, the raw materials, and all other pertinent materials, e.g., intermediate materials and byproducts. With every the input in place algorithm SSG is executed recursively by systematically and combinatorially selecting a series of active sets and carrying out decision mappings. The procedure is terminated when the active sets are exhausted.

Any of the general branch-and-bound methods is inefficient in solving the MIP model of process synthesis: it gives rise to an unduly large number of variables. These methods do not exploit the structural features of the process system. This deficiency is magnified when the model is based on the conventional superstructure containing all possible networks, the majority of which tends to be combinatorial infeasible and thus redundant for any sizeable process.

In contrast, the accelerated branch-and-bound algorithm (algorithm ABB) judiciously exploits the structural features of the process to be synthesized, which manifest themselves in the maximal structure consisting of only combinatorially feasible networks or flowsheets (Friedler et al, 1996). The procedure is initiated at the final or desired products, and proceeds upward through the maximal structure towards the raw materials, i.e., feeds. At each branching step, a decision is made as to which operating unit or units should produce a given material. Algorithm ABB examines if the selection of an operating unit requires an additional operating unit to be selected.

2. PNS Solutions framework

The applicabilities of P-graphs have been explored for nearly 20 years (Heckl et al., 2007; Klemeš and Pierucci, 2008). Following the initial publication of the P-graph methodology (Friedler et al., 1992), it has been extended gradually. Different variations of algorithms MSG, SSG, and ABB have been implemented individually (Friedler, 2010). As a result, a variety of PNS-related programs have been brought to light. Although, there has been no single framework encompassing all the known algorithms. PNS Solutions is designed to rectify this situation. It involves a suitable environment with well defined protocol to which various P-graph-based algorithms and user interfaces can be integrated.

2.1 New software architecture

PNS Solutions contains: (i) the previously developed PNS algorithms as a complete solver program; (ii) the implementation of some new research in this area; and (iii) a graphical user interface (GUI) where the input can be conveniently edited and the output can be displayed. PNS Solutions defines standards and protocols which are useful for its future extension.

In any of the available PNS software, the solver and graphical user interface constitutes a single program; thus, the running time has been highly dependent on the performance of the user's personal computer.
The graphical user interface has two major functionalities: it facilitates the definition of the problem parameters and the visualization of the initial or the resulting P-graph. If the GUI is sharply separated from the solver engine, they may run independent of each other. This also implies that the two parts can be located on different computers after defining an appropriate communication protocol. In software engineering, such architecture is called multi-tiered architecture.

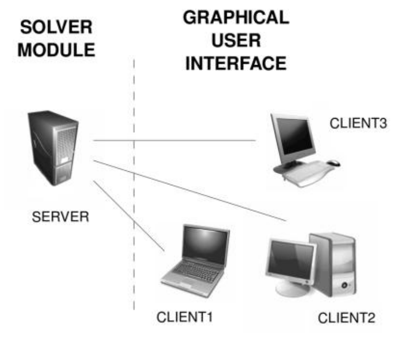


Figure 1: Architecture of the PNS Solutions framework

The simplest multi-tiered architecture comprises two levels, i.e., client and server levels, generally these levels corresponding to the GUI and the solver engine; see Figure 1. The client computer accesses services provided by the solver engine, and reveals it to the user as its own functionality. With an appropriate network connection, the communication in the background, or even the existence of the server, is often transparent to the user. The service provider also known as server is usually a high-performance computer, which deploys its own resources, such as the stored data, the hardware, or other services. The server can use third-party products to provide service without considering further license questions.
The client application is also referred to as client, and the server application server. In the current contribution, the graphical user interface of the new program is considered as a client program and the solver module as a server program. This entails the separation of the visualization and calculation of the application. The client software is a graphical user interface, which facilitates the definition the problem and the visualization of the results.
The server provides a service through which the clients can send problems and can query the solutions. For example, the solution, i.e., output, of algorithm MSG is the maximal structure. The server can manage simultaneously separate clients. Multithread technology is used that ensures parallel communication with the client programs. In addition, the server is capable of parallel processing: the algorithms for various clients run parallel.
In traditional client-server communication, if a client sends to the server a request to run an algorithm that requires a long time to compute, the client will be blocked until the server finishes the run of the algorithm. In our implementation, the communication can

be continued during the solution process of the server, and the client can also access other services on the server.

2.2 Communication within the framework

In information technology, there are a number of well known techniques for network communication. Most of these implement a unidirectional link from a client to a server. Their drawback is that while the solver is functioning, the client must wait in a busy, unresponsive state. In contrast, a bidirectional link renders it possible for the server to contact the client whenever necessary, e.g., when a requested service has been finished. In PNS Solutions, the communication is based on socket (endpoint) links. Between these endpoints, the Transmission Control Protocol (TCP), as a transport protocol, provides a reliable connection. The link between the participants exists continuously, and thus, the security of the communication is high. The server and client programs can be implemented in any, not necessarily the same, programming language, where the TCP sockets are realized. In our framework, a Java-based client is connected to a C++ -based server.

2.3 Solver module

To demonstrate the effectiveness of the PNS Solutions framework, both solver and client modules have been developed and implemented. The solver module contains the majority of the available PNS algorithms and newly developed algorithms, e.g., algorithm GNEG for global neutral extension. PNS Solutions can perform algorithms MSG, SSG, ABB, and “Inside-out”. Algorithms ABB and “Inside-out” have various options to accelerate the solution time. They are the neutral extension, reduced structure generation (RSG), and algorithm GNEG. The current framework facilitates the comparison of the efficiencies of the different reduction procedures.
The solver module can handle different bound values, e.g., lower and upper bounds for the amounts of the materials or the capacity of the operating units. Buying, selling, or penalty prices can be specified for the materials, and fixed and proportional costs are assigned to the operating units.
Beside the optimal solution, it is also possible to generate the first n-best solutions. It is important to provide some degree of freedom for the decision makers because not all the considerations might be involved in the objective function. The solver has been implemented in C++ because speed is particularly important when large scale problems are handled.

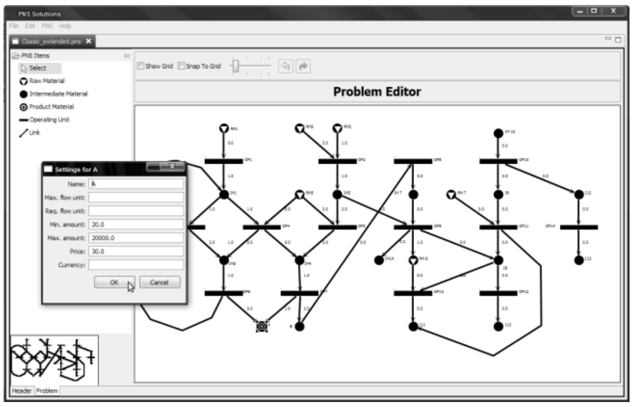
2.4 Graphical user interface

The client program needs to perform a number of tasks. First, it is designed to display the individual problems both graphically and in a text editor form. Second, the client needs to communicate with the server through the network: It sends a problem to the server and requests to run an algorithm. After the server has finished, the client requests the solution. Finally, a standard input and output format has been defined to create a consistence database for examples. The text editor view helps to adjust the system properties. The P-graph of the problem can be established in the visualization area. Various features are available for facilitating the drawing of graphs such as grids, tooltips. These graphs can be exported in various file formats, such as png or eps.

The user interface has been implemented in Java programming language. This has resulted in rapid development as well as portability of the client program to any operating system supporting Java.

3. Integration of the PNS Solutions with instructional modules

The PNS Solutions is also a framework and software with a novel architecture that provides a flexible way to combine different solvers and various graphical editors. As previously discussed, a number of P-graph-based algorithms has been implemented and integrated into this framework as a default server program to demonstrate its efficacy. The default graphical editor is focused on the visual problem editing as it is shown in the figure below.



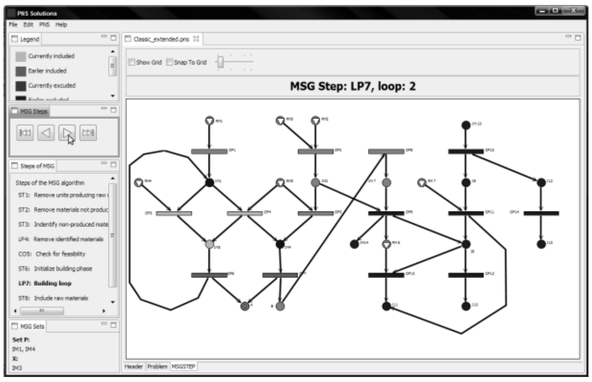
Settings for A

Name:	A
Max. flow unit:	
Req. flow unit:	
Min. amount:	20.0
Max. amount:	20000.0
Price:	30.0
Currency:	

OK Cancel

Figure 2: Editor view

An instructional module has been implemented to provide an interactive demonstration to those who are unfamiliar with the P-graph methodology. For example, a step-by-step version can be executed to view the demonstration of the algorithm MSG; see Figure 3. The server calculates each step, and the user can freely scroll through them. The corresponding notation and legend are provided. Algorithms SSG and ABB are demonstrated similarly: The steps are calculated in advance and buttons are provided to select the next and previous steps of the algorithm.



Steps of the MSG algorithm
ST1: Remove units producing raw
ST2: Remove materials not produc
ST3: Indentify non-produced mate
LP4: Remove identified materials
CO5: Check for feasibility
ST6: Initialize building phase
LP7: Building loop
ST8: Include raw materials

MSG Sets
Set P:
IM1, IM4
X:
IM3

Figure 3: Demonstration of algorithm MSG in runtime

4. Summary

A programming framework has been proposed in the current contribution to facilitate the development and integration of various modules for different PNS problems. The proposed framework is based on the client-server architecture in conjunction with a standardized communication protocol. To illustrate the efficacy of this framework, several algorithms and their user interfaces have been implemented. For instance, the optimal structure can be obtained with both algorithms ABB and “Inside-out”; the former has three versions, which are combinatorially accelerated uniquely and differently. Moreover, each of algorithms MSG, SSG, and ABB is accompanied by a special, tutorial version capable of instructing its operation through stepwise execution and visualization.

References

Biegler L. T., Grossmann I. E. and Westerberg A. W., 1997, Systematic methods of chemical process design, Prentice Hall, USA.

Friedler F., Tarjan K., Huang Y. W. and Fan L. T., 1992, Combinatorial Algorithms for Process Synthesis, Computers & Chemical Engineering 16, S313-S320.

Friedler F., Tarjan K., Huang Y. W. and Fan, L. T., 1993, Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for Maximal Structure Generation, Computers and Chemical Engineering 17, 929-942.

Friedler F., Varga J. B. and Fan L. T., 1995, Decision-Mapping for Design and Synthesis of Chemical Processes: Application to Reactor-Network Synthesis, American Institute of Chemical Engineers Symposium Series (Eds: L. T. Biegler and M. F. Doherty), 91, 246-250.

Friedler F., Varga J. B., Fehér E. and Fan L. T., 1996, Combinatorally Accelerated Branch-and-Bound Method for Solving the MIP Model of Process Network Synthesis, In Nonconvex Optimization and Its Application, State of the Art in Global Optimization, Computational Methods and Applications, Eds: C. A. Floudas and P. M. Pardalos, Kluwer Academic Publishers, USA. 609–626.

Friedler, F., 2010, Process integration, modelling and optimisation for energy saving and pollution reduction, Applied Thermal Engineering, doi: 10.1016/ j.applthermaleng.2010.04.030.

Heckl I., Kovács Z., Friedler F., Fan L. T. and Liu J., 2007, Algorithmic Synthesis of an Optimal Separation Network Comprising Separators of Different Classes, Chemical Engineering and Processing 46, 656-665.

Klemeš J. and Pierucci S., 2008, Emission reduction by process intensification, integration, P–Graphs, Micro CHP, Heat Pumps and Advanced Case Studies, Applied Thermal Engineering 28, 2005-2010.

Pahor B., Kravanja, Z. and Irsic B. N., 2001, Synthesis of reactor networks in overall process flowsheets within the multilevel MINLP approach, Computers and Chemical Engineering 25, 765-774.

Westerberg A. W., 2004, A retrospective on design and process synthesis, Computers and Chemical Engineering 28, 447-458.