Acta Chimica Slovenica 42/1/1995, pp. 15-20

PARALLELIZATION OF THE ACCELERATED BRANCH-AND-BOUND ALGORITHM OF PROCESS SYNTHESIS: APPLICATION IN TOTAL FLOWSHEET SYNTHESIS

J. B. Varga[1,2], F. Friedler[1,2,3], and L. T. Fan[3]

[1]Department of Systems Engineering, Research Institute of Chemical Engineering, Hungarian Academy of Sciences, H-8201 Veszprém, Pf. 125, Hungary

[2]Department of Computer Science, University of Veszprém
Veszprém, H-8200, Hungary

[3]Department of Chemical Engineering, Kansas State University,
Manhattan, KS 66506, U.S.A.

ABSTRACT

That any branch-and-bound algorithm can be easily implemented for parallel processing is remarkable (Eckstein, 1993). Nevertheless, a general-purpose branch-and-bound algorithm is inefficient in solving the mixed-integer nonlinear programming (MINLP) model of process synthesis since an extremely large number of nonlinear programming subproblems is generated, and for each subproblem generated, the number of free variables is exceedingly large; many of these free variables are associated with operating units that must be excluded from any feasible solution of the subproblem.

Combinatorial analyses of both the MINLP models of process synthesis problems and the feasible process structures have yielded mathematical tools for exploiting the unique combinatorial characteristics of process structures in process synthesis. The combinatorially accelerated branch-and-bound algorithm of process synthesis has been constructed by means of these tools. The algorithm gives rise to the optimal solution by minimizing the number of subproblems to be solved and by reducing the size of an individual subproblem. This is accomplished by excluding the binary variables and constraints of those operating units that must not be contained in any feasible solution of the subproblem.

The present work formulates the accelerated branch-and-bound algorithm as a master-slave algorithm for parallel processing. The resultant parallel combinatorially accelerated branch-and-bound algorithm has been rigorously validated by means of combinatorial analysis. Its application to the synthesis of an industrial process indicates that the algorithm is exact effective.

KEYWORDS

Total flowsheet synthesis; parallel algorithm; branch-and-bound algorithm; master-slave approach.

INTRODUCTION

The major source of difficulty in carrying out process synthesis is in its combinatorial nature. For example, if the feasibility of resultant structures is disregarded, the MINLP model of a synthesis problem involving 35 operating units yields $2^{35} \approx 34$ billion combinations, each corresponding to one structure. Solving this problem by a traditional implicit enumeration procedure, e.g., a branch-and-bound

---

Correspondence concerning this paper should be addressed to F. Friedler, Research Institute of Chemical Engineering, Hungarian Academy of Sciences, Veszprém, Pf. 125, 8201, Hungary
E-mail: friedler@almos.vein.hu.

procedure, for the general MINLP model will very likely generate an enormous number of redundant combinations. The number of combinations, however, can be reduced by several orders of magnitude by applying an axiom system expressing the obvious combinatorial properties peculiar to feasible process structures. This axiom system of combinatorially feasible processes reduces the 34 billion combinations of the 35 operating units to 3465 feasible structures, and yet it ensures that the optimal solution is among them. The accelerated branch-and-bound algorithm searches for the optimal structure among these 3465 structures. Since this algorithm divides the problem into highly independent subproblems, it can be effectively rendered parallel.

ACCELERATED BRANCH-AND-BOUND ALGORITHM

A directed bipartite graph termed process graph, or P-graph in short, has been proposed for formal analysis of process structures in process synthesis (Friedler *et al.*, 1992). It is briefly described in the following.

Let M be a given finite nonempty set of materials capable of being transformed in the process of interest. Transformation between two subsets of M is effected in an operating unit of the process. This operating unit must be linked to other operating units through the elements of these two subsets of M. Thus, for the set O of operating units, $O \subseteq \wp(M) \times \wp(M)$. Suppose that (α, β) is an operating unit, i.e., $(\alpha, \beta) \in O$; then, α is the input set, and β is the output set of this operating unit. Pair (M, O) is defined to be a P-graph.

Example 1. Set M_1 of materials and set O_1 of operating units of P-graph (M_1, O_1), illustrated in Fig. 1, are expressed as $M_1 = \{A, B, C, D, E, F, G, H, I, J, K\}$ and $O_1 = \{(\{C, D\}, \{A\}), (\{D\}, \{A, G\}), (\{E\}, \{B, C\}), (\{F, G\}, \{D\}), (\{I\}, \{E\}), (\{J\}, \{F\}), (\{K\}, \{F, H\})\}$.

Let us now induce mapping Δ from M to the set of subsets of set O; in other words, $\Delta[M] \subseteq M \times \wp(O)$. This mapping determines the set of operating units producing material X for any $X \in M$; hence, $\Delta(X) = \{(\alpha,\beta):(\alpha,\beta) \in O \text{ and } X \in \beta\}$.

Definition 1. Let m be a subset of M and X be an element of m, and also let $\delta(X)$ be a subset of $\Delta(X)$ for each $X \in m$. Then, mapping δ from set m to the set of subsets of set O, $\delta[m] = \{(X,\delta(X)):X \in m\}$, is a *decision-mapping* on m.

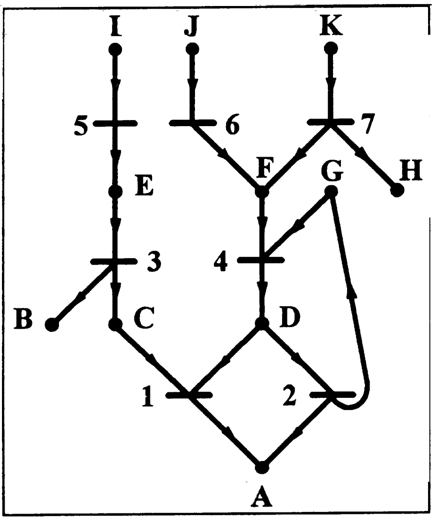

Fig. 1. P-graph of synthesis problem (M_1, O_1).

One of the most important applications of decision-mapping is the acceleration of the conventional branch-and-bound algorithm in solving the MILP or MINLP model of process synthesis. Suppose that set P of the products, set R of the raw materials, and set O of the operating units are known for set M of the materials given. A synthesis problem is defined by triplet (P, R, O); the process structures for this synthesis problem are the subgraphs of P-graph (M, O). The P-graph of a feasible process always needs to be in conformity with certain combinatorial properties, e.g., $P \subset m$, expressed as a set of axioms (Friedler *et al.*, 1992).

The accelerated branch-and-bound algorithm, algorithm ABB, reduces both the number and size of subproblems to be solved by exploiting the combinatorial properties of feasible process structures as expressed in these axioms (Friedler *et al.*, 1991, 1994). The process structures are represented by the decision-mappings in algorithm ABB. This representation ensures consistency and completeness in the system of decisions.

For illustration, the simplified version of algorithm ABB, algorithm ABBS, is given in Fig. 2. Note that procedure BOUND gives a bound for a relaxed subproblem. The possibility of exploiting the parallelism resides in the simultaneous execution of the BOUND procedure.

```
Global variables: R, $\Delta(x)(x \in M)$, U, best;
Comment: procedure BOUND determines a lower bound
for a subproblem;

begin
U:=$\infty$; best := anything; if P=$\emptyset$ then stop;
m:=$\emptyset$; p:=P; ABBS(p, $\emptyset$, $\emptyset$);
if U<$\infty$ then print best
        else print 'there is no solution'
end;

procedure ABBS(p, m, $\delta[m]$):
begin
$x \in p$; C:=$\wp(\Delta(x)) \setminus \{\emptyset\}$;
for all $c \in C$ do begin
      if $\forall y \in m$, $c \cap \overline{\delta}(y) = \emptyset$ & $(\Delta(x) \setminus c) \cap \delta(y) = \emptyset$
      then begin m':=$m \cup \{x\}$;
              $\delta[m'] := \delta[m] \cup \{(x, c)\}$; p':=$p \cup (mat^{in}(c)) \setminus (R \cup m')$;
              if p'=$\emptyset$ then begin U:=min(U, BOUND($\delta[m']$));
                                          update best; end
                                     else if U$\geq$BOUND($\delta[m']$)
                                          then ABBS(p', m', $\delta[m']$);
              end
      end
return
end
```

Fig. 2. Algorithm ABBS.

Example 2. Let us consider the operating units given in Example 1 for producing material A from raw materials I, J, and K. The conventional branch-and-bound algorithm solves 75 subproblems in the worst case to generate the solution; the enumeration tree is given in Fig. 3.

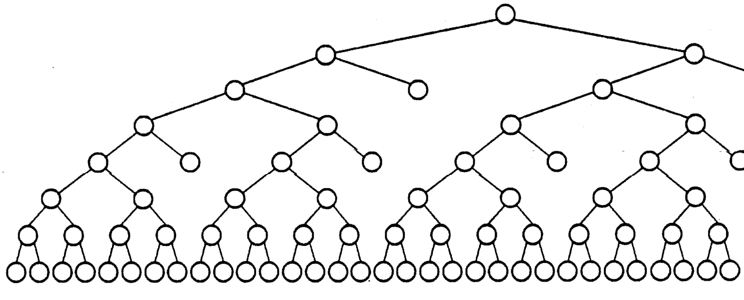

Fig. 3. Enumeration tree for the conventional branch-and-bound algorithm which generates 75 subproblems in the worst case.

For the accelerated branch-and-bound algorithm, the enumeration tree for the same problem shown in Fig. 4 indicates that 9 subproblems are generated in the worst case. An additional reduction is possible by applying some heuristic rules.

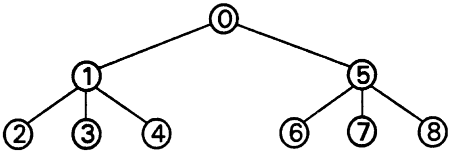

Fig. 4. Enumeration tree for the accelerated branch-and-bound algorithm which generates 9 subproblems in the worst case.

Algorithm ABB preserves the useful property of conventional branch-and-bound of being suitable for parallel implementation.

Example 3. Several subproblems represented in the enumeration tree (see Fig. 4) are independent. For example, after solving subproblem #0, subproblems #1 and #5 can be executed independently, e.g., in parallel; the remaining subproblems are all independent.

"MASTER-SLAVE" ALGORITHMS

An algorithm for the distributed memory system consists of several communicating processes mapped onto several processors; these processes communicate via channels. Depending on the actual mapping of the communicating processes onto the processors, these channels are either virtual or physical links. If two communicating processes are associated with the same processor, they are linked via a virtual channel. If the communicating processes are mapped onto different processors, the corresponding processors must be connected through physical links. Since the word "process" has a different connotation in computer science and chemical engineering, it is eliminated from the terms, master process, slave process, etc. to prevent confusion; instead, simply the terms, master, slave, etc., will be used.

When parallelism is exploited by simultaneously solving independent subproblems or by performing independent tasks, devising an appropriate load balancing strategy is crucial for the most applications. One of the possible approaches to load balancing is to resort to central control, thus giving rise to the so-called master-slave approach where the master controls the activities of the slaves. This approach has several advantages in performing a branch-and-bound algorithm over other load balance strategies. For example, the ratio between the computing and communication activities is usually small for solving a MINLP problem by the branch-and-bound algorithm, i.e., the master can control a large number of slaves. The average size of the subproblems increases with the size of the MINLP problem. Thus, the larger the MINLP problem to be solved the bigger the number of slaves controllable by the master.

The direct connection of each slave to the master would be an ideal configuration; however, the number of available physical links among the processors is constrained and thus, not all the slave can be linked to the master processor. The configuration must satisfy this limitation; moreover, it should be such that unnecessary communications do not significantly affect the speed of executing the algorithm. A simple ring topology is shown in Fig. 5, where the number of available physical links for either of them is 4. Note that a physical link allows a bidirectional communication.

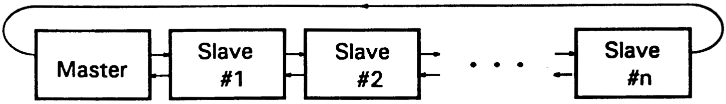

Fig. 5. Simple ring topology.

IMPLEMENTATION OF ALGORITHM ABB ON A TRANSPUTER

Since the number of available physical links is finite, a slave does not simply execute the algorithm; it also must perform various tasks for ensuring the work of the whole system, e.g., passing messages from the master to other slaves. To satisfy this requirement, the slave is decomposed into several communicating processes. The structure of the slave based on the ring topology is shown in Fig. 6. This slave comprises a system of four communicating processes, i.e., the transfer, buffer, algorithm, and multiplexer.

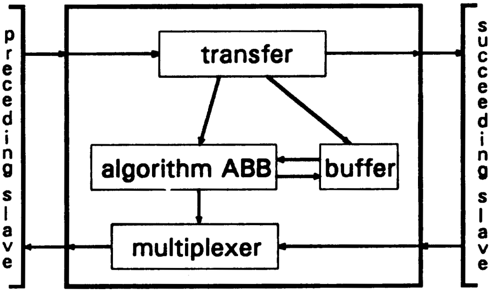

Fig. 6. Structure of an intermediate slave.

The transfer receives messages originating from the master, and depending on its nature, the message is passed to some of the communicating processes including algorithm, buffer, and succeeding slave. The buffer stores the actual bound for the optimal solution of the synthesis problem. The new bounds are received from the transfer and transmitted to the algorithm, if requested. The algorithm performs the task, i.e., solving a subproblem of the branch-and-bound enumeration. The task is received from the transfer, and the result is sent to the master through the multiplexer. The multiplexer simply passes the messages from the algorithm and the other slaves to the master.

The master executes the initial steps, e.g., loading the problem and determining the maximal structure (Friedler *et al.*, 1993). Then, the master controls the slaves' work which includes sending the tasks with the best bound to the idle slaves, collecting the resulting tasks, and sorting them by their bounds; stores the current optimal solution if it exists; and finally detects the termination.

The types of messages involved in the implementation include the following. A "broadcast" message, containing the data necessary for the synthesis problem at the outset of the execution of the algorithm; a "new bound" message to be transmitted to the slaves when a new bound is obtained for the problem; a "task" message containing the necessary data to describe a subproblem; a "solution" message to be sent to the master when a slave obtains a better solution than the current one; a "request" message from a slave to signal to the master that it is idle; and a "signal" message to be transmitted by the master to each slave when the termination is detected.

The implementation mentioned above preserves the advantages of the accelerated branch-and-bound algorithm, i.e., the reduction of both the number and size of the subproblems. It also decreases the execution time of the algorithm almost proportionally to the number of processors.

CONCLUDING REMARKS

A parallel version of the accelerated branch-and-bound algorithm of process synthesis has been developed. It has been realized as a master-slave algorithm on a distributed memory system.

ACKNOWLEDGEMENT

This research was partially supported by the Hungarian Ministry of Education Grant No. 167/94 and Hungarian National Science Foundation Grant No. F016070. Although the research described in this article has been funded in part by the United States Environmental Protection Agency under assistance agreement R-819653 to the Great Plains-Rocky Mountain Hazardous Substance Research Center for U.S. EPA Regions 7 and 8 with headquarters at Kansas State University, it has not been subjected to the Agency's peer and administrative review and, therefore, may not necessarily reflect the views of the Agency. No official endorsement should be inferred this research was also partially supported by the Kansas State University Center for Hazardous Substance Research.

REFERENCES

Eckstein, J. (1993). Parallel branch-and-bound algorithms for general mixed integer programming on the CM-5. Technical report TMC-257, Thinking Machines Corporation, Cambridge, MA.

Friedler, F., K. Tarján, Y. W. Huang, and L. T. Fan (1991). An accelerated branch and bound method for process synthesis. *Fourth World Congress of Chemical Engineering*, Karlsruhe, Germany, 12.2-9.

Friedler, F., K. Tarján, Y. W. Huang, and L. T. Fan (1992). Graph-theoretic approach to process synthesis: axioms and theorems. *Chem. Engng Sci.*, **47**, 1973-1988.

Friedler, F., K. Tarján, Y. W. Huang, and L. T. Fan (1993). Graph-theoretic approach to process synthesis: polynomial algorithm for maximal structure generation. *Computers chem. Engng*, **17**, 929-942.

Friedler, F., J. B. Varga, and L. T. Fan (1994). Decision-Mapping for Design and Synthesis of Chemical Processes: Application to Reactor-Network Synthesis, presented at the International Symposium on Foundations of Computer Aided Process Design, Snowmass Village, CO, U.S.A., July 10-15, (in press).