Using OSL to Improve the Computational Results of a MIP Logistics Model

Lloyd Clarke
Jeff Linderoth
Ellis Johnson
George Nemhauser
The Logistics Institute
School of Industrial & Systems Engineering
Georgia Institute of Technology

Ravi Bhagavan
Michael Jordan
Praxair, Inc.

September 25, 1995

Praxair, Inc. uses a mixed integer programming model to minimize the costs required for the production and distribution of industrial gases across the nation. In conjunction with Georgia Tech's Logistics Engineering Center they have been working on ways to improve the performance (running time and quality of solution) of the model. Praxair, Inc. imposes a 100 minute CPU limit on all computer runs, so finding quality solutions quickly is of critical importance. To date, various customized solution strategies have been implemented in OSL. The purpose of this paper is to describe these rules, their implementation, and their effectiveness.

A historically difficult model for Praxair to solve efficiently is one associated with one of their largest sites. The problem has 1487 rows and 4562 variables, 402 of which are binary. The binary variables make up 234 special ordered sets of type 3. In default mode, OSL is not able to generate an integral solution that is satisfactory to the plants' production planners within the CPU limit.

An effective way to improve the quality of a general integer programming solution strategy is to use knowledge of the underlying physical problem. Our initial efforts concentrated on changing decisions that are currently made by OSL instead of adding additional features like cutting planes. Crucial decisions made by any branch-and bound code are node choice, set choice, and variable choice. OSL has a very good node choice strategy, and there is no simple way to use the problem information to influence this choice. Our work concentrated on good choices for the branching set and variable. In addition, since the goal is to find quality integer solutions quickly, we also investigated the use of a variable fixing heuristic.

The sets of integer variables in the formulation represent a physical plant and a time period. Thus, the branching set can be chosen based on some characteristic of the plant or based on the time period. Three criteria for ranking the importance of the sets of variables were investigated: time, power, and human knowledge. In *time branching*, the first sets of variables upon which to be branched are those corresponding to earliest time periods, regardless of the plant also associated with the set. In *power branching* the first sets of variables upon which to be branched are those corresponding to the plants that require the most power. The production planners at Praxair historically know which plants' configurations are the most vital to the entire system's performance. In *knowledge branching*, the sets of variables are prioritized based on human recommendations. Implementing set choice for *time branching* and *knowledge branching* can be done in OSL with the EKKIMDL routine since it only requires setting priorities for each set of variables prior to beginning branch and bound. Calculation of the branching set is a dynamic process in *power branching* – we must know the linear programming relaxation solution to calculate the power each plant requires. Thus, we use the user exit EKKBRNU.

The variables in the set represent states at which the plant operates. The most important plant configuration decision made by the model is whether or not a plant is to be shut down during a time period. *Shutdown branching* is a scheme that ensures that this is the first decision made for any set of variables. This strategy was implemented by altering the branch point for the chosen set in EKKBRNU.

The *variable fixing heuristic* attempts to find quality feasible solutions quickly by fixing all integral valued binary variables with non-zero reduced cost after the solution of the initial LP. We implemented this scheme directly in the OSL driver by checking and setting the appropriate variables' upper and lower bound values.

To summarize, the main solution strategies and means to accomplish these strategies are shown below:

- Set choice
  - *time branching*
  - *power branching*
  - *knowledge branching*
- Variable choice
  - *shutdown branching*
- *Variable fixing heuristic*

Certain combinations of these strategies are also possible. For example, the *fix-power* branching method first performs the *variable fixing heuristic*, after which *power branching* is performed. The *time-shutdown* branching prioritizes the set of variables as in *time branching* and when branching on these sets of variables, the first decision to be made is whether or not the plant is to be shutdown.

Figure 1 shows a summary of the best solutions obtained within the time limit of 100 minutes by OSL default branching and 5 of the custom strategies. Table 1 shows the number of different solutions and the best solution obtained by many different branching strategies within the CPU limit. The code producing the results was written in IBM XL FORTRAN v2.3, compiled with no code optimization, and run on an IBM RS6000 Model 550. Solution values have been normalized on a scale of zero to one hundred, where zero is the optimal objective function value, and a value of roughly forty is a solution that engineers at Praxair would find acceptable.

Obviously, incorporating knowledge about which plants have the greatest effect on the overall system greatly improves computational effectiveness. In addition, branching first on the variable corresponding to the shutdown configuration within sets of variables representing early time periods has shown to enhance the performance of knowledge-based branching. Variable fixing is able to find quality solutions very quickly and may be of importance on larger instances with similar structure.

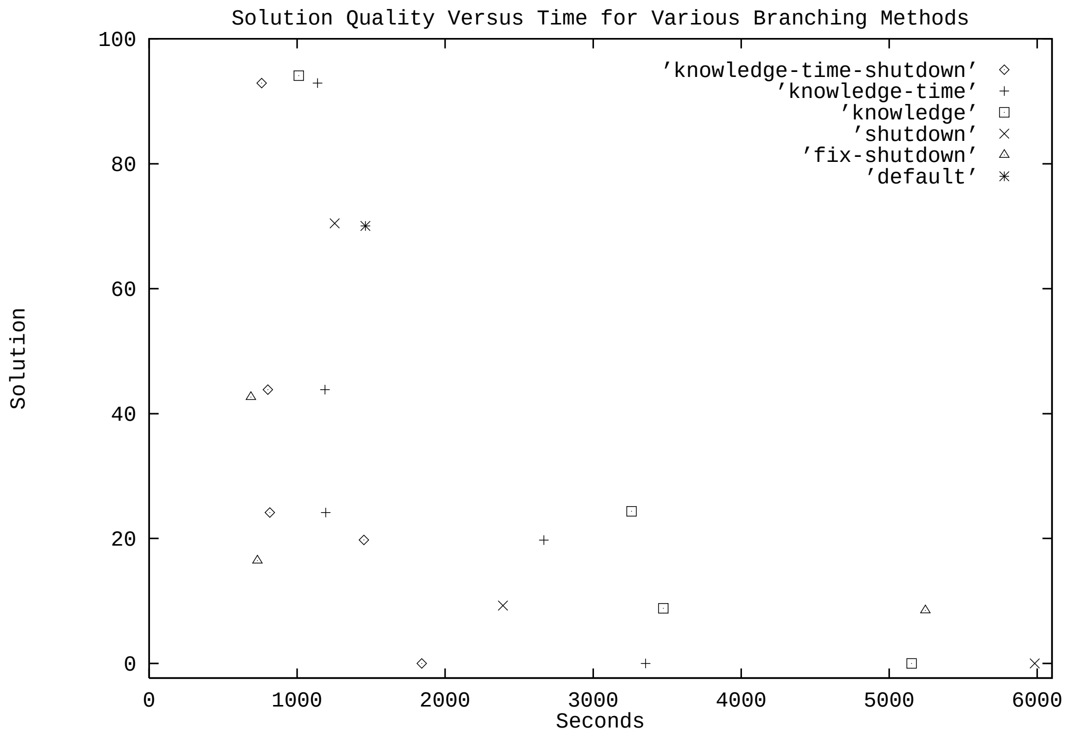

Figure 1: Solution Quality Versus Time.

Strategy	Number of Solutions	Best Solution	Time to First Acceptable Solution	Time to Best Solution
Knowledge-Time-Shutdown	6	0.000	13:35	30:42
Knowledge-Time	6	0.000	19:52	55:53
Knowledge	6	0.000	54:18	1:25:51
Shutdown	5	0.000	39:50	1:39:43
Fix-Shutdown	4	8.473	12:11	1:27:24
Time-Shutdown	4	11.625	12:04	12:19
Fixing	4	18.064	44:29	44:29
Power-Shutdown	3	22.366	1:37:53	1:37:53
Time-Power	3	41.519		4:21
Time	4	42.366		31:39
OSL Default	2	70.037		24:21
Time-Power-Shutdown	2	73.627		12:29
Power	2	89.950		13:56

Table 1: A Comparison of Solution Strategies