November 21, 2002

a. See attachment labeled “Mathematical Formulation.”

b. In order to find the optimal locations of the chlorine booster stations for this portion, I told Excel Solver that my decision variables were binary. The attached pages labeled “Binary Variables, Cmin = 0.25, 0.45, 0.65” show the locations of all of the booster stations, represented by a one in the location variable row near the top of the page. The coverage matrix represents the amount of booster stations at each individual time and injection points are at or above the required chlorine tolerance level. There are also two separate solutions for each Cmin. The first one being the solution taking only the time 1059, 1065, 1071, and 1077 into consideration. The second is the solution taking every time into consideration. The difference between the two time considerations is definitely significant enough to always want to consider all times when doing finding this optimal solution.

c. To find the optimal locations of the chlorine booster stations for this part, I could not explicitly tell Excel Solver that my variables were relaxed, or allowed to vary continuously. I had to specify this manually by use of the constraint function in Excel Solver. To make the decision variables relaxed, I said they had to be greater than or equal to zero for one constraint, and then they had to less than or equal to one in the other constraint. There are also two separate solutions for each Cmin. The format is the same as part b, with different time considerations taken for each Cmin level. As before, the differences are the same. The main conclusion to make from using relaxed variables is this linear programming problem is an integer friendly problem. This is concluded by observing that none of the decision variables’ values change when they are allowed to float from zero to one. This also does not affect the optimal solution as can be seen.

d. For my approximate method of minimizing the amount of chlorine booster stations needed, I used excel to assist my computations. The spreadsheet I used was the "filtered" spreadsheet that was given to us after I made modifications of adding the constraints in the coverage matrix allowing it to do "if/then" statements. An example would be, if I built a booster station at node 1, then will at least one of the nodes 1 - 39 be greater than 1, therefore indicating it being at or above the tolerance level of chlorine. . My first thought was I could do trial and error using the "if/then" statements. This thought was a correct one, but I can cut down on the amount of trials I need to do by hypothesizing where a booster station might need to go.

For the next portion of the explanation, I will use the example where tolerance is set at 0.65. My approximation is based on this observation of the pipeline system: the main pipeline traveling from nodes 1 – 2 – 5 – 6 – 7 – 9 – 11 – 12 – 13 – 14 – 15 – 24 – 23 – 25 – 30 – 26 – 28 are probably not going to need booster stations in them due to their locations being along the main pipeline. The reason the main pipeline doesn't need the booster stations is because there is new water traveling through these nodes constantly, and the newly chlorine boosted water is coming from all directions into the main pipeline. This therefore eliminates much of the trial and error involved. There are also obvious points where a booster stations must be located, which are 3, 4, 8, 10, 18, 19, 21, 29, 33, 35, and 39. The reason this is obvious is because nodes that do not have a lot of water flowing through them are going to need booster stations. So if the node is far from the main pipeline and/or doesn't have many connecting pipes, it will probably need a booster station. To begin the trial and error process, I first set the nodes that obviously needed a booster station to 1 (build booster station at this node). I then checked each of the constraints for each node at each time to see if they were all met. When I saw they were not all above zero, I then chose nodes that looked as if they were also far from the main pipeline and/or didn't have many connecting pipes. I did these one by one to see if the adjacent nodes that were previously zero,

Actually not, due to flow reversal in main line when tank is draining. Network "topology" not a good indicator of behavior.

then turned above one. Another criterion for selecting which number to turn to one was that it had to not meet the constraint, therefore equaling zero. The first node I chose to change to zero was 27. This had a positive impact by making nodes 27, 28, and 34 become greater than or equal to one at all times. Node 31 was next. This merely improved itself to be greater than one, but it had to be done. I now look at the constraints and see that node 37 and one time (1073) at node 17 is not completely supported with chlorine, so I change their values to one. All of the constraints are now met, and the chlorine level at each node is at or above the tolerance level. This method is not exact, but if you would solve this problem exactly, the minimal amount of booster stations would be 14 considering all times of the day. In my approximate solution example, I found there to be a need for 16. Also, the locations of most of the booster stations are the same as in the exact answer.

This example was a specific one, and I felt the problem solving process could be best explained this way. Now knowing the specific example, the algorithm can be stated very simply for any tolerance level. Start from the nodes that are farthest away from the main pipeline. Observe the nodes that need chlorine and move closer to the main pipeline by use of trial and error. It is not a completely accurate way of minimizing the amount of chlorine booster stations needed, but it will get you close as seen in the explained example.

e. There are not many differences between the Integer Linear Programming solution and the relaxed solution. As I stated before, this problem is an integer friendly problem. By allowing the decision variables to float between zero and one doesn't change the optimal solution. The major similarity is that the constraints, other than the ones on the decision variables, are the same in both problem solutions. Also, the optimal solutions and nodes that have booster stations built on them are the same in both solutions.

The approximate solution has slight similarities to the two other solutions. The optimal solutions in the approximate approach are rather close to the Integer and relaxed optimal solutions. In the real world, there would be some error and extra cost in the approximation, but it would not be extreme. Also, the nodes that have booster stations built on them are almost the same. The booster stations are built far from the main pipeline and at points that do not have many connecting pipes. The differences in the approximate solution and the two other solutions are that the approximation is not calculated, but is found by elimination, hypothesizing, and trial and error. The Integer and relaxed solutions are found by Excel solver. The approximation is found by using the constraints set in the coverage matrix as "if/then" statements.

Mathematical Formulation

Objective: Minimize the amount of chlorine booster stations built.

Subject to: - chlorine level at every time must be at or above the tolerance level at every node in the pipeline system

Decision Variables: $x_1 = \{0,1\}$ (no build, build) @ node # 1

$\vdots$

$x_{39} = \{0,1\}$ (no build, build) @ node # 39

$$\text{Minimize } Z = \sum_{i=1}^{39} x_i$$

what is x_{jk}?

Subject to:
$$\boxed{\sum_{\substack{j=1\\k=1}} x_{jk} \geq 1}$$
$j = 1, \ldots, 39$ node
$k = 1, \ldots, 24$ time

Not correct — not all variables included in each constraint.

$$\sum_{j \in S_{ik}} x_j \geq 1$$

S_{ik} = set of all locations such that $C > C_{min}$ at node i & time k.