Paper SAS681-2017

Using SAS/OR® Software to Optimize the Capacity Expansion Plan of a Robust Oil Products Distribution Network

Shahrzad Azizzadeh , SAS Institute Inc.

ABSTRACT

A Middle Eastern company is responsible for daily distribution of over 230 million liters of oil products. For this distribution network, a failure scenario is defined as an occurring when oil transport is interrupted or slows down, and/or when products demands fluctuates outside the normal range. Under all failure scenarios, the company plans to provide additional transport capacity at minimum cost so as to meet all point-to-point product demands. Currently, the company uses a wait-and-see strategy, which carries a high operating cost and depends on the availability of third-party transportation. This paper describes the use of the OPTMODEL procedure to implement a mixed integer programming model to model and solve this problem. Experimental results are provided to demonstrate the utility of this approach. It was discovered that larger instances of the problem, with greater numbers of potential failure scenarios, can become computationally extensive. In order to efficiently handle such instances of the problem, we have also implemented a Benders decomposition algorithm in PROC OPTMODEL.

INTRODUCTION

The Oil Products Distribution Network under study (OPDN) is a network responsible for daily distribution of over 230 million liters of oil products. The network consists of more than 200 consumption regions, 86 oil products warehouses, and several refineries and import/export points that together form the distribution model. After the crude oil is refined, the refined products are pumped through a network of pipelines and sent to major warehouses. The products are then distributed and stored locally to be delivered to retailers and consumers. The part of the distribution process that involves OPDN is where the refined products are delivered to the warehouses from the refineries.

This distribution network faces several possible failure scenarios, which are defined as interruption or slowing down of the products transport, due to loss of transportation capacity in some areas and/or fluctuation of the demand outside the normal range. The company is planning to avoid the possibility of service outage or a system failure by installing additional capacities on links. This would guarantee satisfying the demand under the set of possible failure scenarios.

PROBLEM STATEMENT

One of the preventive measures that supply chain planners take to avoid the possibility of service outage is to install additional capacities on links to guarantee satisfying the demand under the set of possible failure scenarios. Different failure scenarios are characterized by the availability of the links as well as changes to point-to-point demand values for each scenario. The goal is that the resulting network (after capacity expansion decisions are implemented) should be resilient to failures-that is, the network should be able to satisfy the demand under the set of possible failure scenarios, while the total cost of capacity expansions is minimized.

COMPONENTS OF THE PROBLEM

Refineries Refineries provide the products to meet the demands. Each refinery has a large depot where the refined products are kept temporarily before being sent to warehouses via pipelines and railroad tanks.

Warehouses There are 86 local warehouses (including airport fuel depots) in the OPDN. They receive the oil products from the refineries, store them, and finally sell and distribute them to various consumers and retailers, including power plants and gas stations. There are more than 200 consumption regions in the distribution model. Each has been assigned to one or more warehouses.

Means of Transportation Available means of transportation of oil products in the OPDN are pipelines of various diameters and railroad tanks. Although there are scheduling and operational challenges in using pipelines (the need for pumping stations, product flow sequencing, and low speed), they are the most economical means for transporting large quantities of oil products over long distances.

Expansion Options A set of capacity expansion options is available for each of the links to provide additional capacity. This set includes pipelines of various diameters and also railroad tanks. Each of these capacity expansion alternatives has an associated capacity and cost of installation (per kilometer).

Failure Scenarios A set of one hundred potential failure scenarios was identified for the network. The scenarios include partial failures of a subset of links as well as fluctuations in point-to-point demands. An example of a failure scenario is when railroad tanks cannot be used due to freezing conditions, and hence the transportation capacity is reduced, while in the same regions kerosene is being used for heating purposes, and hence the demand increases.

These scenarios are all meaningful in the context of the problem. That is, if any of them happens, given the existing capacities on the links of the network, the demands cannot be satisfied. At the same time, there exists a set of capacity expansion options for the links that, if installed, enables the demands to be satisfied.

Decisions The decisions in the context of the problem are to select an expansion option for each link in the network.

MODELING THE PROBLEM

The proposed modeling approach is general and in principle can be used to address capacity expansion decisions in resilient networks for any type of network, including telecommunication, distribution, and transportation networks. The goal of the problem is to identify the minimum cost capacity expansion plan that can satisfy the demand under the set of failure scenarios. In practice, the capacity expansion decisions needs to be made before any failure scenario occurs, while a second set of decisions need to be made after information about the occurrence of a failure scenario is revealed. The second set of decisions would identify how to route traffic flow on network paths based on the demand and available capacities under a failure scenario. This decision making structure suggests a framework for a two-stage decision-making model, that includes only first stage cost function (capacity expansion investments). The second stage cost is zero, implying that no cost is incurred for routing the flows in the networks once enough capacity is available.

OBJECTIVE

The objective is to minimize the total cost of capacity expansion decisions.

$$Min \sum_{t=1}^{m} \sum_{l=1}^{|L_t|} c_{tl}\ z_{tl}$$

Where m is the number of the links, $|L_t|$ is the size of the set of capacity expansion options available for link e_t, and c_{tl} is the cost of expansion option l on link e_t. The binary variable z_{tl} is 1 if the capacity expansion option l is chosen to be installed on link e_t, and is 0 otherwise. The objective function only includes capacity expansion variables. This implies that as mentioned before, the only cost element in the problem is the cost of capacity expansion.

CONSTRAINTS

Mutual Exclusion Constraints: This set of constraints restricts the choice of capacity expansion options to no more than one for each link.

$$\sum_{l=1}^{|L_t|} z_{tl} \leq 1 \qquad t = 1 \;\; to \;\; m$$

Link Capacity Constraints: The second set of constraints are link capacity constraints. The total amount of flow passing through each link ($\sum_{\forall i,j \in V: d_{ij}>0} \sum_{f=1}^{|P_{ij}|} k_{ijf}^{t} x_{ijf}^{r}$) should be less than or equal to the link's available capacity $\lambda_{tr}\left(u_{t0} + \sum_{l=1}^{|L_t|} u_{tl} z_{tl}\right)$, for all links, under all failure scenarios.

$$\sum_{\forall i,j \in V: d_{ij}>0} \sum_{f=1}^{|P_{ij}|} k_{ijf}^{t} x_{ijf}^{r} \leq \lambda_{tr} \left(u_{t0} + \sum_{l=1}^{|L_t|} u_{tl} z_{tl} \right) \qquad t = 1 \;\; to \;\; m \qquad r = 0 \;\; to \;\; q$$

Where x_{ijf}^{r} represents the second stage decisions, and that is the amount of flow that should be passed through the f^{th} path that connects node i to node j under the scenario r. λ_{tr} is the fraction of the capacity of the link indexed by t under the scenario r.

Demand Constraints: The third set of constraints are demand constraints. The total amount of flow passing through the paths between i and j in scenario r ($\sum_{f=1}^{|P_{ij}|} x_{ijf}^{r}$) should be greater than or equal to the demand between i and j in scenario r (d_{ijr}).

$$\sum_{f=1}^{|P_{ij}|} x_{ijf}^{r} \geq d_{ijr} \qquad \forall \{i,j \in V : d_{ij0} > 0\} \qquad r = 0 \;\; to \;\; q$$

After the variables, numbers, and sets are defined and the data files are read into the procedure, the OPTMODEL code segment in Figure 1 defines the objective and constraints of the problem.

RESULTS

To better examine our modeling and solution approach, we created 12 instances of the problem by incrementally adding to the complexity of the problem. This was done by increasing the number of nodes with demands and incrementally adding the scenarios to generate instances with 51, 79, 85, and 100 scenarios. To solve the problems we used the Mixed Integer Linear Programming (MILP) solver with the default Branch and Cut algorithm in PROC OPTMODEL. We set a time limit of 2 hours for solving the problems. The results are presented in Table 1.

The optimal expansion plan for Instance 1 is presented in Table 2. The network has a total of 442 links, out of which 10 are identifed that require capacity expansion. The optimal value of the objective function is 1,678.2 millions of unit cost for this instance. This amount represents the minimum cost to keep the distribution network functioning under the defined failure scenarios.

```
var x {Paths,Scenarios} >=0   ;

var z {Links_Options}binary ;

min f =sum{<ori,des,opt> in Links_Options} OptionCost[ori,des,opt]*
z[ori,des,opt];

Con MutualExclusion{<ori,des> in links}: sum{opt in options}
z[ori,des,opt]<=1 ;

Con ArcsCapacities{<ori,des> in links, <sce> in scenarios }:
LinkCoef[ori,des,sce]*(Capacity [ori,des]+sum{<opt> in options}
OptionCapacity[ori,des,opt]*z[ori,des,opt])-sum{m in Paths}kfact[ori,des,m]*
x[m,sce]>=0;

Con MeetDemand {i in Nodes,j in Nodes, <sce> in scenarios:
Demand[i,j]>=1E-6}: sum {k in Paths : P_Origin[k]=i
and P_Destination[k]=j } x[k,sce]-DemandCoef[i,j,sce]*demand[i,j]>=0 ;
```

Figure 1: Defining Objective and Constraints in PROC OPTMODEL

Table 1: Execution times

Instance	Execution Time(Seconds)
Inst.1	31.9
Inst.2	82.6
Inst.3	96.6
Inst.4	142.5
Inst.5	190.9
Inst.6	276.0
Inst.7	1,859.4
Inst.8	3,352.4
Inst.9	Exceeded 2 hours
Inst.10	Exceeded 2 hours
Inst.11	Exceeded 2 hours
Inst.12	Exceeded 2 hours

Table 2: Optimal Expansion Plan for Instance 1 Obtained by MILP solver in PROC OPTMODEL

Origin	Destination	Expansion Option
Haan	Maer	Pipe04
Torkan	Maer	Pipe04
Qo	Kan	Pipe36
Latan	Shz	Pipe36
Qan	Ka	Pipe48
Ar	Qo	Pipe48
Maad1	Neaboor	Pipe48
Ten	Qo	Pipe48
Bind	Ken	Pipe48
Sin	Barabbas	Pipe48

ALTERNATIVE SOLUTION APPROACH, BENDERS DECOMPOSITION

The results of our computational experiments show that we are able to solve small to mid-size instances of the problem of finding the optimal capacity expansion plan in a resilient network using the Mixed Integer Linear Programming solver in PROC OPTMODEL, with the branch and cut algorithm. However, we are not able to solve larger size instances of the problem within reasonable time and/or memory resources. This was the motive for designing an alternative way of finding the optimal solution for the problem in PROC OPTMODEL. Designing a solution approach that takes a reasonable amount of time to find the solution would also make it possible to solve similar problems in other industries, including telecommunications, that typically deal with much larger and denser networks.

In order to be able to solve larger instances, we used SAS/OR® PROC OPTMODEL to implement a delayed constraint generation method, also known as the Benders decomposition, that we will discuss in the next section.

DELAYED CONSTRAINTS GENERATION IN PROC OPTMODEL

In order to be able to solve the larger instances of the problem, we implemented a delayed constraint generation method in PROC OPTMODEL. This procedure is also known as the Benders decomposition method. Benders' decomposition is applicable to LP and MILP problems that can be partitioned with respect to the variables. In the multicommodity network design, it has been used for modular network design in [1] and many more.

From our original model, we can observe that z is the vector of complicating variables in the sense that this problem becomes a much easier optimization problem in x when z is temporarily fixed. Particularly, for fixed values of z the problem separates into a number of independent optimization problems (one for each failure scenario), each with a well known network structure, for which the efficient algorithm of Network Simplex (NS) is available in PROC OPTMODEL.

The delayed constraint generation method is an iterative algorithm. The idea behind it is to try to first solve a relaxation of the problem that contains a subset of the variables and constraints (called Relaxed Master Problem or RMP), and to check afterwards whether any of the relaxed constraints are being violated by solving a set of smaller problems, called sub-problems. If any of these constraints are violated with a specific value of $z = \bar{z}$, then an associated feasibility cut is constructed and added to the RMP at the current iteration. The new RMP is solved again, and the process iterates until all of the sub-problems become feasible. Here, what we chose to form the initial RMP is the objective function and the set of mutual exclusion constraints. The rest of the constraints are separable by failure scenario and form linear and independent sub-problems. Each sub-problem contains the demand constraints and link capacity constraints associated with a failure scenario.

At each iteration, using the current value of z variables obtained by solving the current RMP, the feasibility of each sub-problem is tested. For this purpose we have added a dummy variable w as the objective and to the constraints as the extra capacity on the links. If the min value of w is greater than 0, then the interpretation is that the sub-problem is infeasible and an associated cut should be added. Here, we have implemented a multi-cut [2] approach. That is, in every iteration, rather than iterating and re-solving a new RMP as soon as one constraint is violated, all of the violated constraints are obtained at the same time and the associated feasibility cuts are constructed. The algorithm then iterates, and a new RMP is constructed by adding all the required cuts. It is then solved again. This is done because it potentially decreases the number of times the RMP problem, which is an integer programming problem and hence hard- is solved.

Since accessing the dual values of constraints is easy in PROC OPTMODEL, generating cuts and implementing such an algorithm is straight forward. As well, we use a COFOR loop in PROC OPTMODEL to solve sub-problems concurrently.

The code segment in Figure 2 shows how the sub-problems are solved and for the constraints that are being violated the dual values are retrieved.

SUMMARY OF RESULTS The summary of the results is presented in Table 3. We are able to solve all of our 12 instances using the Benders decomposition method. For smaller

```
min f_sub{i in s}= w[i];
problem p{i in S} include f_sub[i] w[i]
{j in Paths} xp[j,i]
{<ori,des> in links} ArcsCapacities [ori,des , i]
{k1 in Nodes,k2 in Nodes : Demand[k1,k2]>=1e-6} MeetDemand [k1 ,k2 , i ]  ;

cofor{i in S} do;
      use problem p[i];
        solve with lp/algorithm=ns;
 for {<ori,des> in links} do;
 if w[i]>1e-6 then
 DualCap[ori,des,i]=ArcsCapacities[ori,des,i].dual;
 end;

 for {k1 in Nodes,k2 in Nodes : Demand[k1,k2]>=1e-6} do;
  if w[i]>1e-6 then
 DualDem[k1,k2,i]=MeetDemand[k1,k2,i].dual;
 end;
end;
```

Figure 2: PROC OPTMODEL code segment showing how sub-problems are solved concurrently and dual values are retrieved.

Table 3: Comparing execution times by solving directly and through Benders' decomposition

	Execution Time (Seconds)	
Instance	Direct	Decomposition
Inst.1	31.9	30.6
Inst.2	82.6	105.2
Inst.3	96.6	110.1
Inst.4	142.5	130.5
Inst.5	190.9	55.3
Inst.6	276.0	142.7
Inst.7	1,859.4	155.0
Inst.8	3,352.4	185.6
Inst.9	Exceeded 2 hours	208.8
Inst.10	Exceeded 2 hours	684.0
Inst.11	Exceeded 2 hours	626.2
Inst.12	Exceeded 2 hours	758.0

instances, the execution time is relatively small, and the time or memory requirement do not grow rapidly as the size of the instances grow, as it does when using the MILP solver directly.

CONCLUSIONS

In this paper we presented a solution approach that uses SAS/OR® to address a minimum cost capacity expansion problem in a robust oil products distribution network. The MILP model finds an expansion option for each link, such that all the demands are satisfied under the set of potential failure scenarios, while the overall cost is minimized. Two set of decisions are made in this process. In the first set, an expansion option is selected for each link (or it is decided that no expansion is required). In the second set, given the expansion decisions, the model decides how to route the flow of products on network paths under each failure scenario. The proposed modeling and solution approach can be extended and applied to other types of networks, including telecommunications, distribution, and transportation networks.

REFERENCES

1. Geir Dahl and Mechthild Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS J. on Computing*, 10(1):111, January 1998.

2. Franois Louveaux John R. Birge. *Introduction to Stochastic Programming.* Springer New York, 2011.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.
Contact the author at:

Shahrzad Azizzadeh
C2272, 820 SAS Campus Drive, Cary NC 27513 USA
Phone: (919) 531-6266
Shahrzad.Azizzadeh@sas.com
www.sas.com