Principal Investigator/Project Director: Dr. Roger Z. Ríos
Institution: Universidad Autónoma de Nuevo León (Mexico)
Award Number: CONACYT J33187-A **Program:** CONACYT Young Investigator Program
Project Title: "Intelligent Optimization of Natural Gas Transmission Networks"

On NLP and MINLP Formulations and Preprocessing for Fuel Cost Minimization of Natural Gas Transmission Networks

Yanet Villalobos-Morales
Diana Cobos-Zaleta
Humberto J. Flores-Villarreal
Conrado Borraz-Sánchez
Roger Z. Ríos-Mercado

Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León
AP 111-F, Cd. Universitaria
San Nicolás de los Garza, NL 66450
México
{yanet, diana, humberto, conrado, roger}@yalma.fime.uanl.mx

ABSTRACT

The problem of minimizing fuel consumption on natural gas pipeline networks is addressed. Both a nonlinear programming model and a mixed-integer nonlinear programming model are presented. A database containing many problem instances under different types of topologies is proposed and described. For a more efficient application of optimization algorithms, preprocessing techniques for this problem are presented, discussed, and computationally evaluated. It is found the use of three techniques provides a significant algorithm performance improvement reducing considerably many of the numerical difficulties inherent to this very complex problem. In addition, a preliminary computational evaluation of an outer approximation with equality relaxation and augmented penalty algorithm for MINLPs is presented. In initial findings, the algorithm reports promising results by finding optimal solutions to many problem instances.

1. INTRODUCTION

Natural gas is transported by pressure throughout a pipeline system. This transmission produces energy loss caused by the existing friction between the gas and the pipeline's inner wall, and by the heat transfer between the gas and the environment. Compressor stations are installed in the network to increase the pressure level and keep the gas moving. Typically, compressor stations consume in fuel about 3 to 5% of the total gas transported through the network. The problem of finding out how to operate the stations becomes significantly important due to the high amount of gas transported daily through the system.

There are several variations of this problem depending on the modeling assumptions and the decisions to be made. In this work we describe two models. The first is a model where we consider two types of decision variables: mass flow rate through each arc and pressure value at each node (Wu et al., 2000). Both are continuous variables so the model is a nonlinear programming (NLP) problem. We assume that it is known in advance how many compressor units are operating within each compressor stations. The second model is an extension of the first, where discrete decision variables that represent the number of compressor units to be turned on are introduced. In this case, the NLP becomes a mixed-integer nonlinear program (MINLP), where the discrete variables are precisely the number of compressor units to be used within the station. Both models are nonconvex, which make them hard to solve.

In this paper, we state the main assumptions and describe a NLP and a MINLP models. We present and propose a database of problem instances that we use in our work. Due to numerical difficulties found when attempting to use a generalized reduced gradient (GRG) algorithm to solve this problem, we present and describe some preprocessing techniques. These are computationally evaluated over a variety of problem instances. We found the application of these techniques yield significant algorithm performance improvement. In addition, we present a preliminary computational evaluation of an outer approximation with equality relaxation and augmented penalty algorithm (Viswanathan and Grossmann, 1990) for the MINLP model. The results are promising as we were able to find optimal solutions for several problem instances. We end up this paper by highlighting our current and future work in this project.

2. MATHEMATICAL MODEL

2.1 Modeling Assumptions

In the present paper, we make the following modeling assumptions.

- We assume that the problem is in steady state. This is, our model will provide solution for systems that have been operating for a relative large amount of time. Transient analysis would require increasing the number of variables and the complexity of this problem.

- The network is balanced. This means that the sum of all the net flows in each node of the network is equal to zero. In other words, the total supply flow is driven completely to the total demand flow without loss. We know that compressor stations are feed with some of the fuel driven by the pipelines. For sustaining the zero mass balance assumption, we consider the cost of this consumption as an extra opportunity cost that represent the amount we would spend if we were to buy the fuel from third parties.
- Each arc in the network has a pre-specified direction.
- Each parameter is known (deterministic).

2.2 The NLP Model

Parameters:

V:	Set of all nodes in the network
Vs:	Set of supply nodes ($Vs \subseteq V$)
Vd:	Set of demand nodes ($Vd \subseteq V$)
Ap:	Set of pipeline arcs
Ac:	Set of compressor station arcs
A:	Set of all arcs in the network; $A = Ap \cup Ac$
U_{ij}:	Arc capacity of pipeline (i,j); $(i,j) \in Ap$
R_{ij}:	Resistance of pipeline (i,j); $(i,j) \in Ap$
P_i^L, P_i^U:	Pressure lower and upper limit at each node; $i \in V$
B_i:	Net mass flow rate at node i; $i \in N$. $B_i > 0$ if $i \in Vs$, $B_i < 0$ if $i \in Vd$, $B_i = 0$ otherwise

Variables:

x_{ij}:	Mass flow rate in arc (i,j);); $(i,j) \in A$
p_i:	pressure at node i; $i \in V$

Formulation:

$$\text{Minimize} \quad \Sigma_{(i,j)\in Ac} \; g_{(i,j)}(x_{ij}, p_i, p_j) \tag{1a}$$

$$\Sigma_{\{j|(i,j)\in A\}} x_{ij} - \Sigma_{\{j|(I,j)\in A\}} x_{ji} = B_i \qquad i \in V \tag{1b}$$

$$X_{ij} \leq U_{ij} \qquad (i,j) \in Ap \tag{1c}$$

$$p_i^2 - p_j^2 = R_{ij}\, x_{ij}^2 \qquad (i,j) \in Ap \tag{1d}$$

$$p_i^L \leq p_j \leq p_i^U \qquad i \in V \tag{1e}$$

$$(x_{ij}, p_i, p_j) \in D_{ij} \qquad (i,j) \in Ac \tag{1f}$$

$$x_{ij}, p_i \geq 0 \tag{1g}$$

Constraints (1b)-(1c) are the typical network flow constraints representing node mass balance and arc capacity, respectively, where $\sum_{i\in V} B_i = 0$. Equation (1d) respresents the gas flow dynamics in each pipeline of the network in steady state. Equation (1e) denotes the limits of pressure in each node. In constraint (1f), D_{ij} represents the feasible operating domain for compressor station (i,j).

For a single centrifugal compressor unit (i,j), its operating domain D_{ij}, as a function of the variables x_{ij} (flow through the arc (i,j)), p_i (inlet pressure) an p_j (outlet pressure), is given by the following set of equations.

$$\frac{h_{ij}}{s_{ij}^2} = A_H + B_H\left(\frac{q_{ij}}{s_{ij}}\right) + C_H\left(\frac{q_{ij}}{s_{ij}}\right)^2 + D_H\left(\frac{q_{ij}}{s_{ij}}\right)^3$$

$$S_{ij}^{\ L} \le S_{ij} \le S_{ij}^{\ U}$$

$$R^L \le \frac{q_{ij}}{s_{ij}} \le R^U$$

Here the variables (h_{ij}, q_{ij}, s_{ij}) are the adiabatic head, volumetric flow and speed of the compressor and are related to (x_{ij}, p_i, p_j) by the following equations:

$$h_{ij} = \frac{ZRT_s}{m}\left[\left(\frac{p_i}{p_j}\right)^m - 1\right],$$

$$q_{ij} = ZRT_s \frac{x_{ij}}{p_i},$$

where the following parameters are assumed to be known with certainty:

A_H, B_H, C_H, D_H	Constants, which depend on the type of compressor (typically estimated by the least squares method).
T_s	Gas temperature
Z	Gas compressibility factor
R	Gas constant
M	=(k-1)/k, where k is the specific ratio
S^L	Speed lower bound
S^U	Speed upper bound
R^L	Surge (lower limit of q_{ij} / s_{ij})
R^U	Stonewall (upper limit of q_{ij} / s_{ij})

The following auxiliary variables are introduced:

q_{ij}	Inlet volumetric flow rate in compressor (i,j); (i,j) $\in$ Ac
h_{ij}	Adiabatic head compressor (i,j); (i,j) $\in$ Ac
s_{ij}	Compressor speed.

Physically, the operator directly knows how to set up the compressor in terms of the variables h_{ij}, q_{ij}, and s_{ij}; however, given the mapping from (h_{ij}, q_{ij}, s_{ij}) to (x_{ij}, p_i, p_j), it is preferable to work on the later from the network optimization perspective because mass flow rate (x_{ij}) is observed at every node. Figure 1 illustrates this domain in the (x_{ij}, p_i, p_j) space for a fixed value of x_{ij}.

For a detailed explanation about centrifugal compressor station and previous work, see Ríos-Mercado (2002).

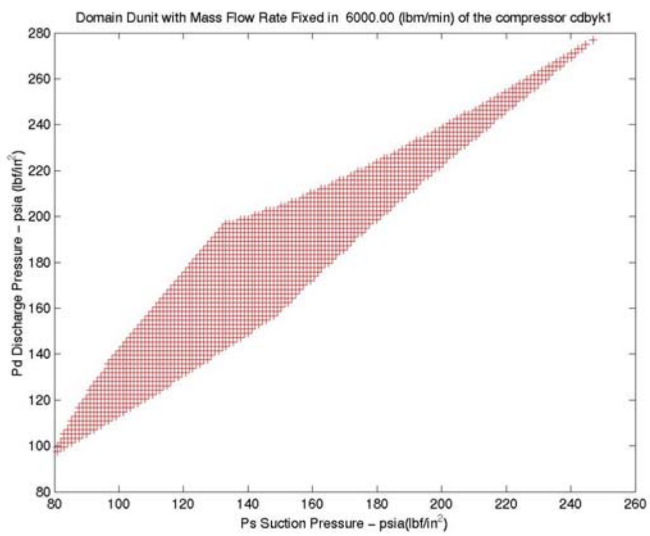

Figure 1. Domain of a compressor unit with x_{ij} fixed at 6000 lbm/min

As we can appreciate, from Figure 1, the domain of a centrifugal compressor is non-convex. Besides, it is well known that the behavior of each compressor is non-linear. Furthermore, the feasible domain in (1d) is a non-convex set and the objective function is also non-convex. These features make this problem particularly nasty.

2.3 Extension to a MINLP Model

In the previous section, a NLP formulation was presented under the assumption that the number of compressor units to be operating within in each station is known in advance. Let us now assume that this is not known, so this number of individual units is treated as a variable. We introduce the following:

N_{ij}: Upper bound on the number of compressor units in station (i,j); $(i,j) \in Ac$

n_{ij}: Integer variable that denotes the number of compressor units to be operating at station (i,j); $(i,j) \in Ac$

Then, assuming the compressor units are all identical and hooked up in parallel within each station (see Fig. 2), constraints (1f) are replaced by (2) and (3).

$$\left(\frac{x_{ij}}{n_{ij}}, p_i, p_j\right) \in D_{ij} \qquad (i,j) \in Ac \qquad (2)$$

$$n_{ij} \in \{0,1,2,\ldots,N_{ij}\} \qquad (i,j) \in Ac \qquad (3)$$

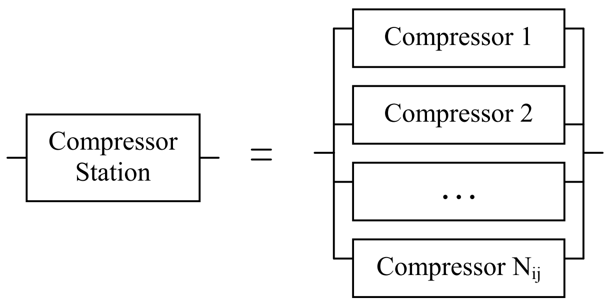

Figure 2. Representation of a compressor station

It is important to highlight that the mass flow rate that pass through a single centrifugal compressor unit, when considering n_{ij} identical units within the station, can be equally split into the number of centrifugal compressors working at each compressor station. The flow trough each unit becomes x_{ij}/n_{ij} so $(x_{ij}/n_{ij}, p_i, p_j)$ must satisfy the feasible operating domain for a single compressor unit D_{ij} as represented in (2). A more detailed description can be found in Wu (1998).

In addition, the total fuel consumed at station (i,j) is given by:

$$g_{(i,j)}(x_{ij}, p_i, p_j, n_{ij}) = n_{ij}\, g_{(i,j)}\left(\frac{x}{n}, p_i, p_j\right) \qquad (4)$$

where $g_{(i,j)}(x_{ij}, p_i, p_j)$ is the fuel used by a single unit. So the objective (1a) is replaced by (4). The new MINLP model becomes to minimize (4) subect to (1b)-(1e), (1g), (2) - (3).

There are some algorithms for solving MINLPs (Floudas, 1995) such as the outer approximation with equality relaxation and penalty augmented (Viswanathan and Grossmann, 1990) that only allow for binary discrete variables. In this case, the model would have to be modified in the following way. A binary variable n_{ijk} which is equal to one if the k-th compressor of compressor station (i,j) is working, and 0 otherwise is introduced. Then we add the equation $\sum_k x_{ijk} = n_{ij}$ $\forall (i,j) \in Ac$; and allow n_{ij} to become a real variable.

3. Database Design

The purpose of designing and seting up a database with problem instances is twofold. First, it is necessary for testing our proposed algorithms. Second, it will provide a more efficient and reliable test for benchmarking different algorithms. As far as we know, there is no such database for this type of problems. So one of our contributions is to design and build this database, which we now describe.

The first step in setting up a database application is to define what the system network must accomplish. The database will be very important for the scientific and technological activity in our research.

To define our database we need to specify:

1. *The system input.* The information that will be entered into the computer.
2. *The system processing functions.* The calculations the computer must carry out and the ways information is to be transferred among database files.
3. *The system output.* The way in which the data will be displayed on the screen and printed on files.

There are three different kinds of topologies: (a) simple or gun-barrel (Figure 3), (b) tree (Figure 4), and (c) cyclic (Figure 5). Characteristics of the database: It is comprised of 17 different topologies (3 type A, 7 type B, and 7 type C). There are 2 more topologies taken from real-world pipeline companies in Louisiana and Texas.

Figure 3 shows an example of an instance definition. All these instances are available at: **http://yalma.fime.uanl.mx/~pisis/** .

In Figures 4 and 5, a green node (shown with an incoming arrow next to it) represent a supply point, a red node (shown with an outgoing arrow next to it) represents a demand points, and a white node is a transshipment node. A pipeline is represent by a single directed arc joining two nodes, and a compressor is represented by a boxed directed arc joining two nodes.

```
//Short Network without loop, of 6 nodes and 2 compressor

gasoductos:
1       2       50      3   0.0085
3       4       50      3   0.0085
5       6       50      3   0.0085

compresor:
2       3       TIPO-9  1
4       5       TIPO-9  1

propiedades:
0.95    0.6248  519.67  1.287

nodos:
1       600     800     600
2       600     800     0
3       600     800     0
4       600     800     0
5       600     800     0
6       600     800     -600
```

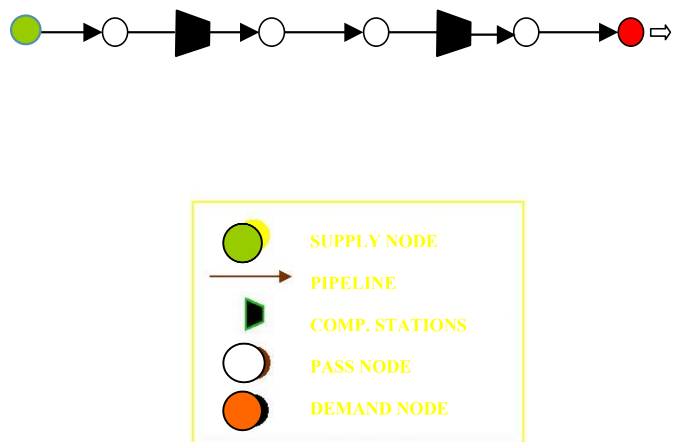

Figure 3. Example of a simple topology.

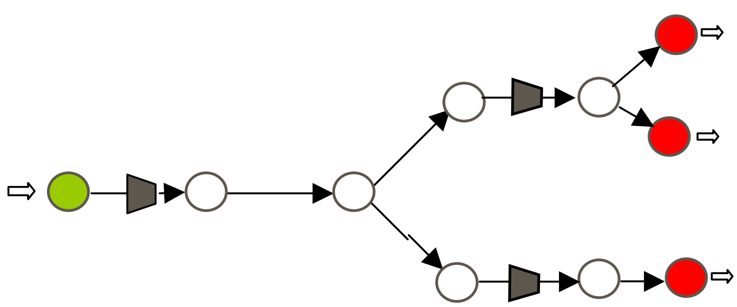

Figure 3. Example of a tree topology.

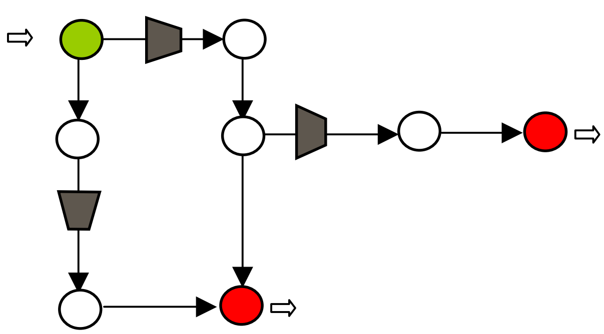

Figure 4. Example of a cyclic topology.

4. PREPROCESSING

When we first attempted to use a generalized reduced gradient (GRG) algorithm (Bazaraa, Sherali, and Shetty, 1993) for solving the NLP model, we noticed that practically all instances tried (either the NLP or the MINLP model) (were not solved due to numerical difficulties of many kinds. This motivated our work in preprocessing techniques.

In short, pre-processing techniques can be defined, in a general way, as elementary operations that transform a problem formulation into an equivalent model that may lead to better algorithmic properties before attempting to solve it. By applying the pre-processing techniques, one expects that the optimization algorithm avoids many of these numerical difficulties.

There are many preprocessing techniques available in the literature. Among these, we have used the following three:

- **Variable bounding:** This is done to reduce the feasible region or search space, preventing the algorithm to examine boundedless domains. This typically reduces the computational effort.

- **Initial variable value assignment:** It is known that an optimization algorithm such as gradient search method moves from a feasible solution to another feasible solution. So starting from a feasible solution or as close to a feasible solution as possible helps in reaching a feasible point more quickly, and hence may lead to a better algorithmic performance. In this vein, we have observed that initial default values make the algorithm perform poorly in our problem. So, to deal with this we sort to assigning initial values to all variables selecting a value within variable bounds

- **Scaling:** This is intended to have all coefficients in each constraint with the same or similar order of magnitude. This technique becomes important because optimization algorithms work internally with matrices. When matrices entries differ greatly in magnitude, many numerical difficulties can be introduced making the execution to fail or to report a non-feasible solution where there is one indeed. Scaling involves a careful selection of units for variables.

.

5. COMPUTATIONAL EVALUATION

5.1 Evaluation of Preprocessing Techniques

The first part of this experiment was to evaluate the impact of using the preprocessing techniques (discussed in Section 4) in this problem. In order to do this, we constructed the model by using

GAMS (Brooke, Kendrick, and Meeraus, 1992) and use a GRG built-in optimizer called CONOPT (Drud, 1992). This was implemented in a Sun Ultra 10 under Solaris 7 OS.

We apply the method under different scenarios (depending on the preprocessing techniques used) on the three different kinds of topologies (as depicted in Figures 4, 5, and 6) using different compressor types. For each compressor type, many instanvces using different flow values were tested. In scenario E1, variable bounding is used. In scenario E2, both variable bounding ad initial value assignment are used. In scenario E3, in addition to E2, scaling is used as well, so all three pre-processing techniques are applied.

Type of compressor	**Number of instances tested**	**Number of local optimal solutions**		
		E1	**E2**	**E3**
Snarlin-k1	12	11	12	12
Rakeey-k1	10	10	10	10
Rakeey-k2	17	14	17	16
Hamper-k1	19	12	16	17
Bellvan-k1	10	10	10	10
Bellvan-k2	10	10	10	10
Bellvan-k3	17	12	14	17
Bethany-k1	18	14	14	13
Bethany-k2	16	12	15	15
Total	**129**	**105**	**118**	**120**

Table 1. Comparison of scenarios in the in simple topologies.

Type of compressor	**Number of instances tested**	**Number of local optimal solutions**		
		E1	**E2**	**E3**
Snarlin-k1	17	17	17	18
Rakeey-k1	15	14	15	15
Rakeey-k2	16	7	8	8
Hamper-k1	11	1	1	3
Bellvan-k1	9	9	9	9
Bellvan-k2	9	9	9	9
Bellvan-k3	17	9	14	14
Bethany-k1	12	8	9	9
Bethany-k2	5	5	5	5
Total	**112**	**79**	**87**	**90**

Table 2. Comparison of scenarios in tree topologies.

Type of compressor	Number of instances tested	Number of local optimal solutions		
		E1	E2	E3
Snarlin-k1	23	21	23	23
Rakeey-k1	19	18	19	19
Rakeey-k2	25	20	23	25
Hamper-k1	31	15	16	20
Bellvan-k1	15	15	15	15
Bellvan-k2	15	15	15	15
Bellvan-k3	22	22	22	22
Bethany-k1	20	18	16	17
Bethany-k2	19	19	18	17
Total	**189**	**163**	**167**	**173**

Table 3. Comparison of scenarios in cyclic topologies.

The results are shown in Tables 1, 2, and 3. In each row, it is shown the number of feasible or local optimum solutions found under scenario. As can be seen, the results obtained applying the three preprocessing techniques produced very encouraging results in the three topologies. In most of the instances tested, local optima was reached.

When comparing the number of iterations used by the algorithm we can see how E3 (using all preprocessing techniques) is a better choice in terms of number of local optima found. This was verified statically by performing a non-parametric test which was significant at a level α = 95%. Detailed results and tests can be found in Villalobos-Morales (2002).

5.2 Solving the MINLP Model: Preeliminary Results

The purpose of the second part at the experiment was to obtain a preliminary computational experience and finding out how difficult was to obtain feasible solutions and/or local optima in a MINLP model, in order to gain insight into the problem and to later devise a better way to solve this problem.

To do this, we implemented the model in GAMS. To solve the MINLP we use DICOPT (GAMS Development Corporation, 2000), which is a built-in outer approximation with equality relaxation method (Viswanathan and Grossmann, 1990). We only consider a simple topology (Figure 4), which consists of 6 nodes (one demand, one supply), and 5 arcs (2 compressors and 3 pipelines). For this topology, 9 different instances (with different compressor type each) with data taken from real-world units, were tested. The model was run on a Sun Ultra 10 under Solaris 7 OS.

We attempted to solve each of the nine MINLP instances with a flow value of 950 MMCFD and applied a preprocessing phase, which consisted of scaling some of the constraints. The results are shown in Table 4. The compressor type is shown in the first column. The model status column indicates the stopping criteria used by DICOPT, where "Intermediate non integer" means that the solver failed in the MINLP sub-problem but found a solution feasible to the NLP relaxation, "Integer solution" means that the solver was able to find a feasible solution, "Locally optimal" means that a local optimal solution was found, and "Non integer solution" means that the solver has a failure and it could not find a feasible solution. The third column shows the numerical value of the objective function that represents the fuel consumption cost. The fourth and fifth columns shows the number of iterations and CPU time (sec.), respectively. The last two columns show the CPU time (sec.) and percentage taken by each sub-problem (CPLEX was used for the MIP subproblem and CONOPT for the NLP subproblem).

As we can see, the algorithm found optimal or feasible solution in 5 of 9 instances. This illustrates the importance of an appropriate scaling and preprocessing phase. But it also shows that further work is necessary at pre-processing to derive models with no numerical difficulties. For the instances solved, we can also observe that most of the time was spent on solving the MINLP sub-problem.

Type of Compressor	Model status	Objective function	Number of iterations	CPU time (sec.)	CPLEX (time / %)	CONOPT2 (time / %)
Bellvan-k1	Integer solution	1892084.9	463	1.99	0.06 / 3.02	1.93 / 96.98
Bellvan-k2	Integer solution	1892084.9	463	2.054	0.08 / 3.9	1.97 / 96.1
Bellvan-k3	Integer solution	3664136.4	123	0.439	0.03 / 6.84	0.45 / 97.83
Bethany-k1	Non integer solution	1139777.3	91	0.461	0.01 / 2.17	0.45 / 95.73
Bethany-k2	Integer solution	4569340.9	123	0.469	0.02 / 4.27	0.45 / 95.75
Snarlin-k1	Locally optimal	979896.9	54	0.079	0.02 / 25.37	0.06 / 74.63
Rakeey-k1	Locally optimal	756923.9	30	0.085	0 / 0	0.09 / 100
Rakeey-k2	Integer solution	1396385.2	188	0.849	0.05 / 5.89	0.8 / 94.11
Hamper-k1	Locally optimal	1683290.2	3	0.021	0 / 0	0.02 / 100

Table 4. Results of experimentation made on the MINLP model.

6. CONCLUSIONS

It is evident the tremendous positive impact that the preprocessing techniques had in the problem addressed. The application of those techniques not only found local optima but also reduced the resources (number of iterations) used by the computer. This was statically confirmed by using non-parametric tests, which were all significant and conclude that E3 (using all three techniques) was the scenario where the best performance was observed in this type of problem. Results of this work have been published (Villalobos-Morales, 2002).

As far as solving NLP is concerned, we are now in the process of evaluating and comparing algorithms such as GRG and Lagrangian projection. Our goal is to find the best algorithm parameters for which best local optima can be found.

Other issue for further investigation is the choice of a good starting point. For the experiments, we set the initial point as the average of the upper and lower bound, for all variables. However as we have learned from this work, many variables end up with values far away from these initial ones. So, this suggests that other choices for setting these initials values may help getting feasible solutions more quickly, and thus faster convergence to local optima can be obtained.

As far as solving the MINLP problem is concerned, we have observed how preprocessing can also be very helpful. We have started obtaining local optimal for one type of problem instances (simple topology), but we also observed that NLP solvers still experience in some cases numerical difficulties in making progress toward meeting all optimality conditions. Further work is under way now to attempt to exploit the current problem structure se we can deal with these difficulties successfully.

This is an ongoing research. We are still working on preprocessing to address the numerical difficulties obtained when applying the DICOPT algorithm for solving the MINLP. We are also in the process of evaluating more MINLP instances that will give us a more complete picture and a better understanding for solving this very important problem more effectively. All programs and files used in this work are available from the authors or at the web page: **http://yalma.fime.uanl.mx/~pisis/** .

Acknowledgments: This research is supported by the Mexican National Council for Science and Technology (CONACYT grant J33187-A).

REFERENCES

M. S. BAZARAA, H. D. SHERALI, AND C. M. SHETTY (1993). *Nonlinear Programming: Theory and Algorithms.* Wiley, New York, USA.

A. BROOKE, D. KENDRICK, AND A. MEERAUS (1992). *GAMS. A User's Guide, Release 2.25.* Scientific Press, San Francisco.

A. DRUD (1992). CONOPT: A large scale GRG code. *ORSA Journal on Computing,* 6(2): 207-216.

C. A. FLOUDAS (1995). *Nonlinear and Mixed-lnteger Optimization: Fundamentals and Applications.* Oxford University Press, New York.

GAMS DEVELOPMENT CORPORATION (2000). *GAMS: The Solver Manuals.* Washington, DC.

R. Z. RÍOS-MERCADO (2002). Natural gas pipeline optimization. In P. M. Pardalos and M. G. C. Resende, editors, *Handbook of Applied Optimization,* Chapter 18.8.3, 813-825. Oxford University Press, New York.

Y. VILLALOBOS-MORALES (2002). *Pre-Procesamiento de un Problema de Optimización de Redes de Gas Natural.* Master thesis, Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, México. Translation: Pre-processing of a natural gas optimization problem

Y. VILLALOBOS-MORALES AND R. Z. RÍOS-MERCADO (2002). Approximating the fuel consumption function on natural gas centrifugal compressors. In *Proceedings of the 2002 NSF Design, Service, and Manufacturing Grantees Research Conference,* San Juan, Puerto Rico.

J. VISWANATHAN AND I. E. GROSSMANN (1990). A combined penalty function and outer approximation method for MINLP optimization. *Computers and Chemical Engineering*, 14:769-782.

S. WU (1998). *Steady-State Simulation and Fuel Cost Minimization of Gas Pipeline Networks,* Ph.D. dissertation, University of Houston, Houston.

S. WU, R. Z. RÍOS-MERCADO, E. A. BOYD, AND L. R. SCOTT (2000). Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. *Mathematical and Computer Modeling,* 31(2-3): 197-220.